INTRODUCCIÓN

Este documento es de fundamental ayuda para todo aquel profesional en el área que requiera conocimientos específicos y netamente técnicos acerca de ***AUTORIDAD CERTIFICADORA OPEN SOURCE (CAOS) .***
Se recomienda leerlo previamente y en su totalidad antes de cualquier soporte, cambio, reingeniería o proceso alguno que facilite o personalice el funcionamiento de la aplicación.

Se ofrece toda la información, especificaciones, requerimientos y una guía total y completa de todo el software que involucra el desarrollo, la implementación, el funcionamiento del sistema desarrollado y mencionado en esta parte.

CAPITULO 1

1.1 DIAGRAMA DE CLASES

En este diagrama se mostrara el funcionamiento del sistema como tal.

Su representación esta dada por el siguiente formato

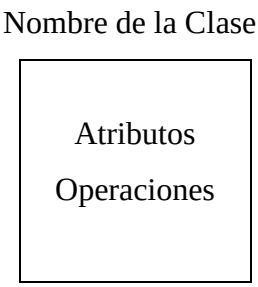

Figura 1.1

1.1.1 Diagrama de clases para la Unidad Certificadora.

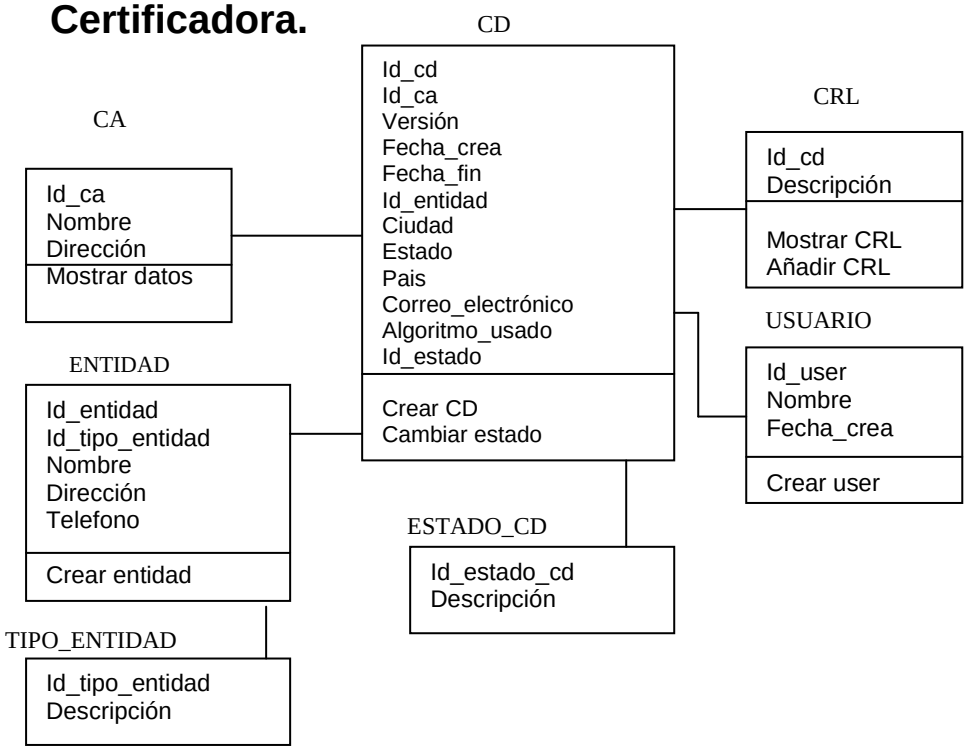

Figura 1.2

1.1.2 Diagrama de clases para la Consulta de notas

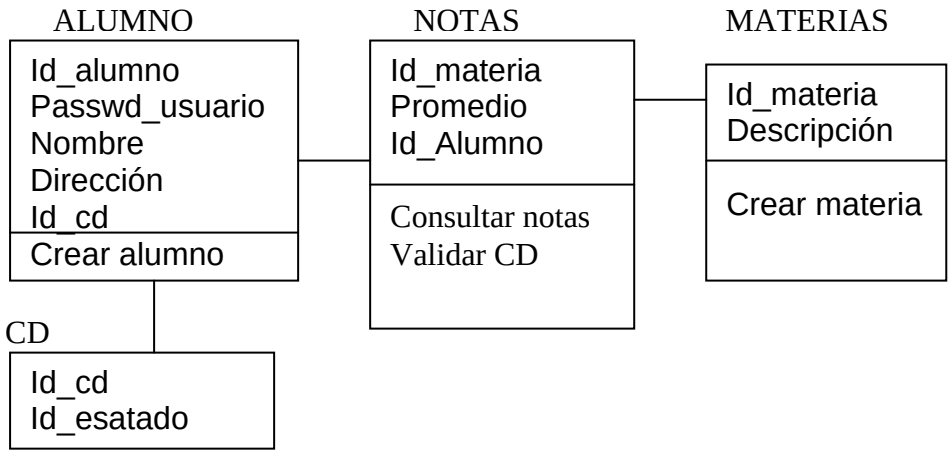

Figura 1.3

1.2 DISEÑO DE LA BASE DE DATOS

1.2.1 Estándares de La Base de Datos

1.2.1.1 Estándar de la base de datos para la Unidad Certificadora.

En la base de datos de la unidad de Certificación utilizaremos los estándares que definiremos a continuación.

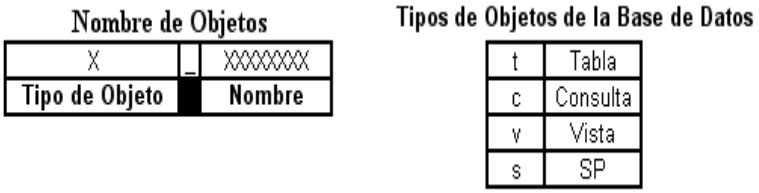

Figura 1.4

Ejemplo: t_usuario

Figura 1.5

Ejemplo: t_usuario_nombre

1.2.1.2 Estándar de la base de datos para la consulta de notas.

En la base de datos de notas utilizaremos los estándares que definiremos a continuación con la finalidad de tener un orden y mayor facilidad de identificar los diferentes elementos que la componen.

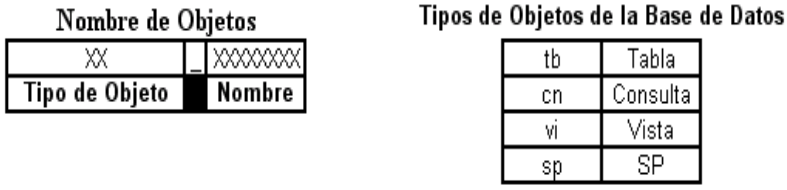

Figura 1.6

Ejemplo: tb_alumno cn_notas

Figura 1.7

Ejemplo: tb_alumno_codigo cn_materia_nombre

1.2.2 Definición de Entidades

Para efectos del desarrollo CA se han definido las siguientes entidades.

CA	USUARIO
CD (Certificado digital)	ENTIDAD

Para efectos del desarrollo de la base de datos para la consulta de notas se han definido las siguientes entidades.

Alumno

Materia

1.2.3 Atributos por Entidad

De las entidades antes definidas utilizaremos los siguientes atributos:

1.2.4 Atributos para la Unidad Certificadora

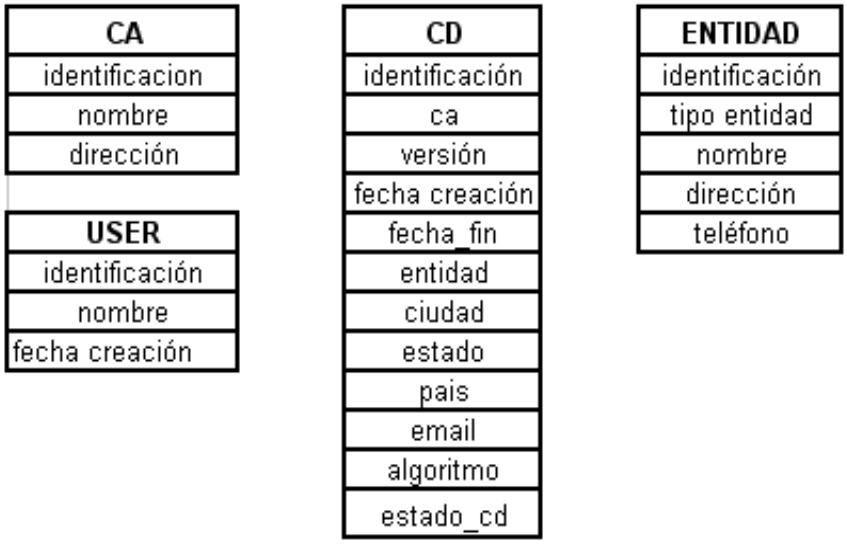

Figura 1.8

1.2.5 Atributos para la consulta de notas.

Figura 1.9

1.3 Diagrama Entidad Relación

1.3.1 DER para la Unidad Certificadora

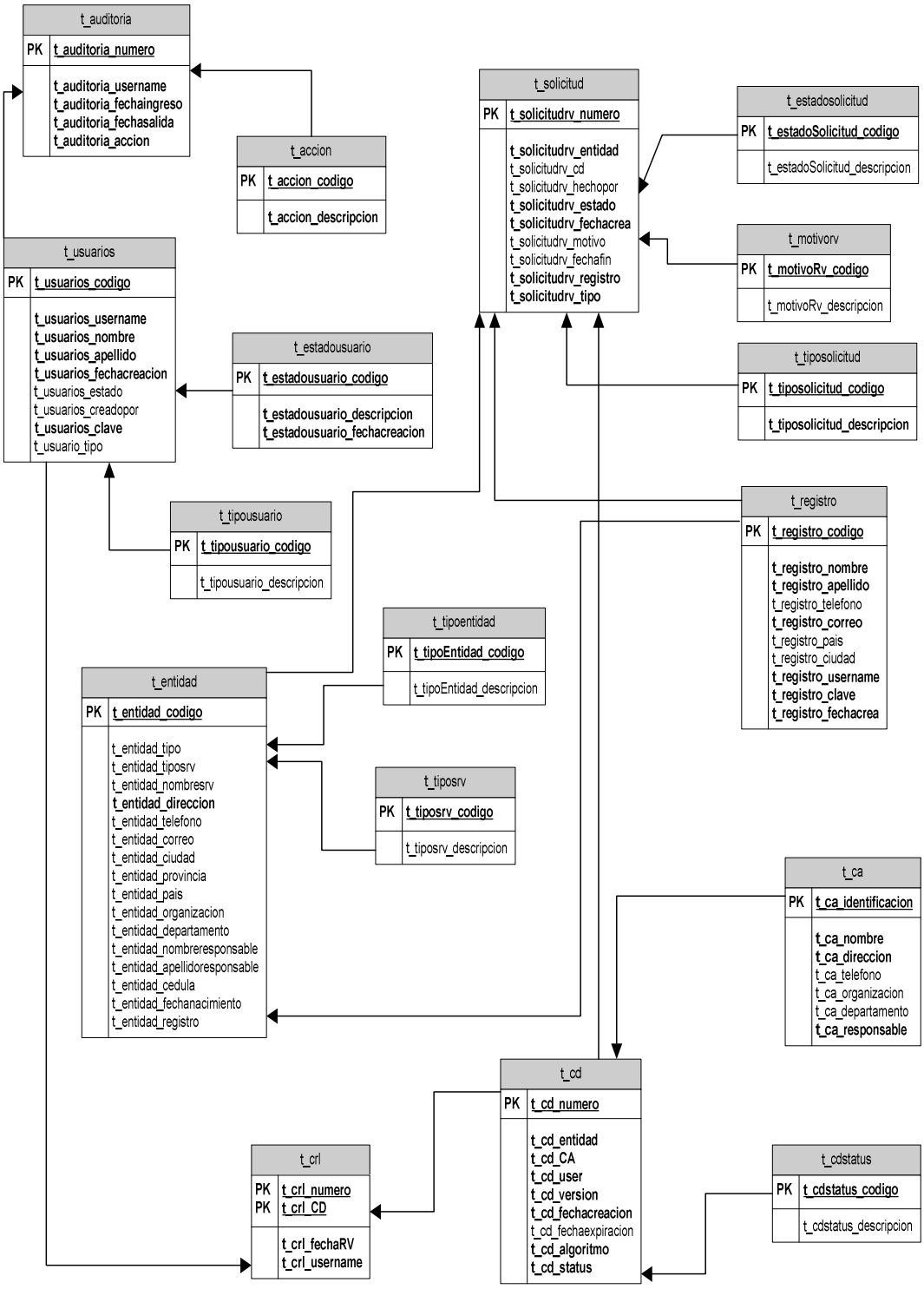

Figura 1.10

1.3.2 DER para la consulta de notas

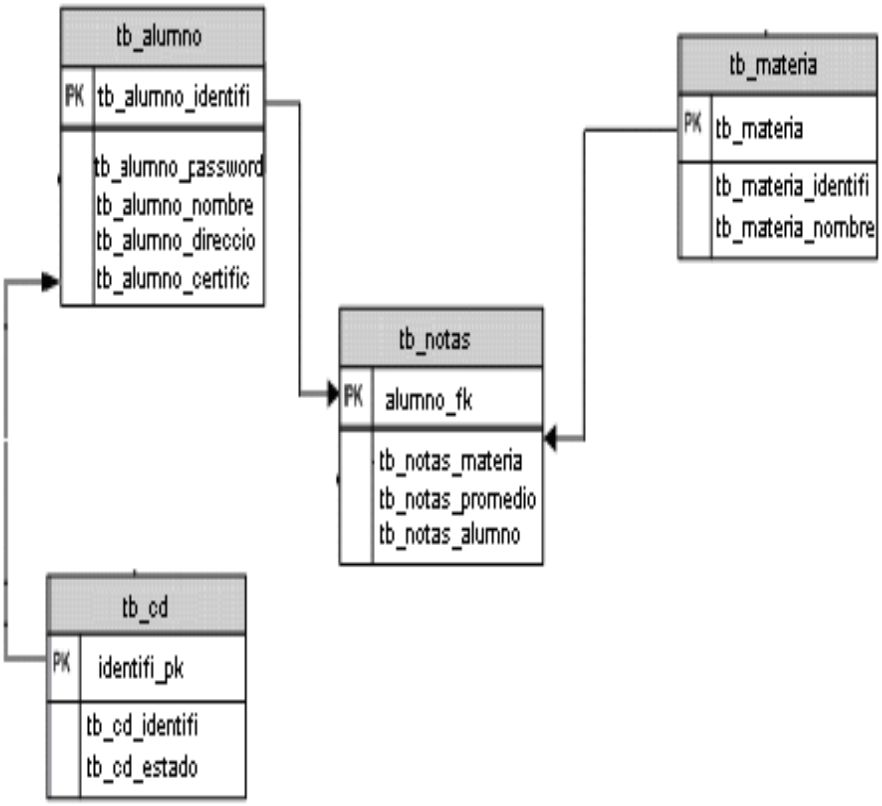

Figura 1.11

CAPÍTULO 2

2.1 DICCIONARIO DE DATOS

2.1.1. Diccionario de datos para la Unidad Certificadora

El diccionario de datos a utilizarse está detallado en el capitulo 3 del Tomo I

UNIVERSIDAD DE GUAYAQUIL Facultad de Ciencias Matemáticas y Físicas Carrera Ingeniería en Sitemas		DICCIONARIO DE DATOS	Página 1 de Fecha de 15/12/2006
AUTORES: - Katherine Medina - Diana Vinueza - Manuel Durán	PROYECTO DE GRADUACION	SISTEMA PARA UNIDAD CERTIFICADORA RAIZ	

NOMBRE DE LA TABLA	TIPO TABLA	LONGITUD DEL REGISTRO	MEDIO DE	MEDIO DE
t_ca	maestra		Disco Duro	- Disco 3 1/2 " - Cd Writer

DESCRIPCIÓN: Esta tabla contendrá la informacion del C. A.

DESCRIPCION DEL REGISTRO

No.	CAMPO	DESCRIPCIÓN	TIPO	FORMATO		RANGOS VALIDOS	REGLAS DE VALIDACIÓN
1	t_ca_identificacion	Codigo Identificación del C. A.	PK	N	I		Obligatorio
2	t_ca_nombre	Nombre de la C.A.	D	VC			Obligatorio
3	t_ca_direccio	Direccion de la C. A.	D	VC			Obligatorio
4	t_ca_telefono	Telefono de la CA	D	VC			
5	t_ca_organizacion	Nombre de la Organización	D	VC			
6	t_ca_departamento	Departamento responsable de la CA	D	VC			
7	t_ca_responsable	Nombre del responsable de la CA	D	VC			Obligatorio

OBSERVACIÓNES:

TIPO:	FORMATO GENERAL	FORMATO NUMERICO
PK → Clave Primaria FK → Clave Foránea D → Campo descriptivo	N → Numericc C → Char VC → VarChar D → Date	I → Inte S → Smallint De→ Dec M → Money

Figura 2.1

2.1.2. Diccionario de datos para la consulta de notas

El diccionario de datos a utilizarse está detallado en el capitulo 3 del tomo I.

UNIVERSIDAD DE GUAYAQUIL Facultad de Ciencias Matemáticas y Físicas Carrera Ingeniería en Sitemas		DICCIONARIO DE DATOS	Página 1 de Fecha de Elaboración: 12/09/2006
AUTORES: - Katherine Medina - Diana Vinueza - Manuel Durán	PROYECTO DE GRADUACION	SISTEMA PARA UNIDAD CERTIFICADORA RAIZ	

NOMBRE DE LA TABLA	TIPO TABLA	LONGITUD DEL REGISTRO	MEDIO DE	MEDIO DE RESPALDO
tb_notas	Relacion		Disco Duro	- Disco 3 1/2 " - Cd Writer

DESCRIPCIÓN: Esta tabla contendrá la informacion de las notas a mostrar

DESCRIPCION DEL REGISTRO

No.	CAMPO	DESCRIPCIÓN	TIPO	FORMATO		RANGOS VALIDOS	REGLAS DE VALIDACIÓN
1	tb_notas_materia	Identificacion de materia	FK	N	I		Obligatorio
2	tb_notas_promedio	Promedio	D	VC			Obligatorio
3	tb_notas_alumno	identificacion de alumno	FK	N	I		Obligatorio

OBSERVACIÓNES:

TIPO:	FORMATO GENERAL	FORMATO NUMERICO
PK → Clave Primaria FK → Clave Foránea D → Campo descriptivo	N → Numerico C → Char VC → VarChar D → Date	I → Integer S → Smallint De→ Decimal M → Money

Figura 2.2

2.2 Scripts de la creación de Base de Datos

2.2.1 Scripts de la Base Datos Para CAOS

create table `t_auditoria` (
`t_auditoria_numero` int(11) NOT NULL auto_increment,
`t_auditoria_username` int(3) NOT NULL,
`t_auditoria_fechaingreso` datetime NOT NULL,
`t_auditoria_fechasalida` datetime default NULL,
`t_auditoria_accion` int(3) NOT NULL,
PRIMARY KEY (`t_auditoria_numero`),
KEY `FK_t_auditoria` (`t_auditoria_accion`),
KEY `FK_t_auditoriauser` (`t_auditoria_username`),
CONSTRAINT `FK_t_auditoria` FOREIGN KEY (`t_auditoria_accion`)
REFERENCES `t_accion` (`t_accion_codigo`),
CONSTRAINT `FK_t_auditoriauser` FOREIGN KEY (`t_auditoria_username`)
REFERENCES `t_usuarios` (`t_usuarios_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

create table `t_ca` (
`t_ca_identificacion` int(3) NOT NULL auto_increment,
`t_ca_nombre` tinytext NOT NULL,
`t_ca_direccion` tinytext NOT NULL,
`t_ca_telefono` tinytext,
`t_ca_organizacion` tinytext,
`t_ca_departamento` tinytext,
`t_ca_responsable` tinytext NOT NULL,
PRIMARY KEY (`t_ca_identificacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

create table `t_cd` (
`t_cd_numero` int(10) NOT NULL auto_increment,
`t_cd_entidad` int(3) NOT NULL,
`t_cd_CA` int(3) NOT NULL,
`t_cd_user` int(3) NOT NULL,
`t_cd_version` tinytext NOT NULL,
`t_cd_fechacreacion` datetime NOT NULL,
`t_cd_fechaexpiracion` datetime default NULL,
`t_cd_algoritmo` tinytext NOT NULL,
`t_cd_status` int(3) NOT NULL,
`t_cd_publicado` int(3) NOT NULL default '0',
PRIMARY KEY (`t_cd_numero`),
KEY `FK_t_cd_ca` (`t_cd_CA`),
KEY `FK_t_cduser` (`t_cd_user`),
KEY `FK_t_cdestadouser` (`t_cd_status`),
KEY `FK_t_cd` (`t_cd_entidad`),
CONSTRAINT `FK_t_cdestadouser` FOREIGN KEY (`t_cd_status`)
REFERENCES `t_estadousuario` (`t_estadousuario_codigo`),
CONSTRAINT `FK_t_cduser` FOREIGN KEY (`t_cd_user`) REFERENCES
`t_usuarios` (`t_usuarios_codigo`),

```
  CONSTRAINT `FK_t_cd_ca` FOREIGN KEY (`t_cd_CA`) REFERENCES `t_ca`
(`t_ca_identificacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

create table `t_cdstatus` (

```
  `t_cdstatus_codigo` int(3) NOT NULL,
  `t_cdstatus_descripcion` tinytext,
  PRIMARY KEY  (`t_cdstatus_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

create table `t_crl` (

```
  `t_crl_numero` int(3) NOT NULL auto_increment,
  `t_crl_CD` int(10) NOT NULL,
  `t_crl_fechaRV` datetime NOT NULL,
  `t_crl_username` int(3) NOT NULL,
  PRIMARY KEY  (`t_crl_numero`),
  KEY `FK_t_crl` (`t_crl_CD`),
  KEY `FK_t_crlusers` (`t_crl_username`),
  CONSTRAINT `FK_t_crl` FOREIGN KEY (`t_crl_CD`) REFERENCES `t_cd`
(`t_cd_numero`),
  CONSTRAINT `FK_t_crlusers` FOREIGN KEY (`t_crl_username`)
REFERENCES `t_usuarios` (`t_usuarios_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

create table `t_entidad` (

```
  `t_entidad_codigo` int(3) NOT NULL auto_increment,
  `t_entidad_tipo` int(3) default NULL,
  `t_entidad_tiposrv` int(3) default '0',
  `t_entidad_nombresrv` tinytext,
  `t_entidad_direccion` tinytext NOT NULL,
  `t_entidad_telefono` tinytext,
  `t_entidad_correo` tinytext,
  `t_entidad_ciudad` tinytext,
  `t_entidad_provincia` tinytext,
  `t_entidad_pais` tinytext,
  `t_entidad_organizacion` tinytext,
  `t_entidad_departamento` tinytext,
  `t_entidad_nombreresponsable` tinytext,
  `t_entidad_apellidoresponsable` tinytext,
  `t_entidad_cedula` tinytext,
  `t_entidad_fechanacimiento` date default NULL,
  `t_entidad_registro` int(3) default NULL,
  PRIMARY KEY  (`t_entidad_codigo`),
  KEY `FK_t_entidad` (`t_entidad_tipo`),
  KEY `FK_t_entidadregistro` (`t_entidad_registro`),
  KEY `FK_t_entidad4` (`t_entidad_tiposrv`),
  CONSTRAINT `FK_t_entidad` FOREIGN KEY (`t_entidad_tipo`) REFERENCES
`t_tipoentidad` (`t_tipoEntidad_codigo`),
  CONSTRAINT `FK_t_entidad4` FOREIGN KEY (`t_entidad_tiposrv`)
REFERENCES `t_tiposrv` (`t_tiposrv_codigo`),
  CONSTRAINT `FK_t_entidadregistro` FOREIGN KEY (`t_entidad_registro`)
REFERENCES `t_registro` (`t_registro_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
create table `t_estadosolicitud` (
  `t_estadoSolicitud_codigo` int(3) NOT NULL,
  `t_estadoSolicitud_descripcion` tinytext,
  PRIMARY KEY  (`t_estadoSolicitud_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `t_estadousuario` (
  `t_estadousuario_codigo` int(3) NOT NULL,
  `t_estadousuario_descripcion` tinytext NOT NULL,
  `t_estadousuario_fechacreacion` datetime NOT NULL,
  PRIMARY KEY  (`t_estadousuario_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

create table `t_motivorv` (
  `t_motivoRv_codigo` int(3) NOT NULL,
  `t_motivoRv_descripcion` tinytext,
  PRIMARY KEY  (`t_motivoRv_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

create table `t_registro` (
  `t_registro_codigo` int(3) NOT NULL auto_increment,
  `t_registro_nombre` tinytext NOT NULL,
  `t_registro_apellido` tinytext NOT NULL,
  `t_registro_telefono` tinytext,
  `t_registro_correo` tinytext NOT NULL,
  `t_registro_pais` tinytext,
  `t_registro_ciudad` tinytext,
  `t_registro_username` tinytext NOT NULL,
  `t_registro_clave` tinytext NOT NULL,
  `t_registro_fechacrea` datetime NOT NULL,
  PRIMARY KEY  (`t_registro_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

create table `t_solicitud` (
  `t_solicitudrv_numero` int(10) NOT NULL auto_increment,
  `t_solicitudrv_entidad` int(3) NOT NULL,
  `t_solicitudrv_cd` int(10) default NULL,
  `t_solicitudrv_hechopor` tinytext,
  `t_solicitudrv_estado` int(3) NOT NULL,
  `t_solicitudrv_fechacrea` datetime NOT NULL,
  `t_solicitudrv_motivo` int(3) default NULL,
  `t_solicitudrv_fechafin` datetime default NULL,
  `t_solicitudrv_registro` int(3) NOT NULL,
  `t_solicitudrv_tipo` int(3) NOT NULL,
  PRIMARY KEY  (`t_solicitudrv_numero`),
  KEY `FK_t_solicitudrvEntidad` (`t_solicitudrv_entidad`),
  KEY `FK_t_solicitudrvEstado` (`t_solicitudrv_estado`),
  KEY `FK_t_solicitudrvmotivo` (`t_solicitudrv_motivo`),
  KEY `FK_t_solicitudrvregistro` (`t_solicitudrv_registro`),
  KEY `FK_t_solicitudtipo` (`t_solicitudrv_tipo`),
  CONSTRAINT `FK_t_solicitudrvEstado` FOREIGN KEY (`t_solicitudrv_estado`)
REFERENCES `t_estadosolicitud` (`t_estadoSolicitud_codigo`),
  CONSTRAINT `FK_t_solicitudrvmotivo` FOREIGN KEY (`t_solicitudrv_motivo`)
REFERENCES `t_motivorv` (`t_motivoRv_codigo`),
```

```
  CONSTRAINT `FK_t_solicitudrvregistro` FOREIGN KEY
(`t_solicitudrv_registro`) REFERENCES `t_registro` (`t_registro_codigo`),
  CONSTRAINT `FK_t_solicitudtipo` FOREIGN KEY (`t_solicitudrv_tipo`)
REFERENCES `t_tiposolicitud` (`t_tiposolicitud_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
create table `t_tipoentidad` (
  `t_tipoEntidad_codigo` int(3) NOT NULL COMMENT 'Codigo del tipo de
entidad',
  `t_tipoEntidad_descripcion` tinytext,
  PRIMARY KEY  (`t_tipoEntidad_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
create table `t_tiposolicitud` (
  `t_tiposolicitud_codigo` int(3) NOT NULL auto_increment,
  `t_tiposolicitud_descripcion` tinytext NOT NULL,
  PRIMARY KEY  (`t_tiposolicitud_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
create table `t_tiposrv` (
  `t_tiposrv_codigo` int(3) NOT NULL,
  `t_tiposrv_descripcion` char(50) default NULL,
  PRIMARY KEY  (`t_tiposrv_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
create table `t_tipousuario` (
  `t_tipousuario_codigo` int(3) NOT NULL auto_increment,
  `t_tipousuario_descripcion` char(50) default NULL,
  PRIMARY KEY  (`t_tipousuario_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
create table `t_usuarios` (
  `t_usuarios_codigo` int(3) NOT NULL auto_increment,
  `t_usuarios_username` tinytext NOT NULL,
  `t_usuarios_nombre` tinytext NOT NULL,
  `t_usuarios_apellido` tinytext NOT NULL,
  `t_usuarios_fechacreacion` datetime NOT NULL,
  `t_usuarios_estado` int(3) default '1',
  `t_usuarios_creadopor` tinytext,
  `t_usuarios_clave` tinytext NOT NULL,
  `t_usuario_tipo` int(3) default '2',
  PRIMARY KEY  (`t_usuarios_codigo`),
  KEY `FK_t_usuariosestado` (`t_usuarios_estado`),
  KEY `FK_t_usuariostipo` (`t_usuario_tipo`),
  CONSTRAINT `FK_t_usuariosestado` FOREIGN KEY (`t_usuarios_estado`)
REFERENCES `t_estadousuario` (`t_estadousuario_codigo`),
  CONSTRAINT `FK_t_usuariostipo` FOREIGN KEY (`t_usuario_tipo`)
REFERENCES `t_tipousuario` (`t_tipousuario_codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2.2.1 Scripts de la Base Datos Para Consulta de notas

```
create database consulta_notas;
use consulta_notas;
create table tb_cd
( tb_cd_identifi int,
tb_cd_estado char,
constraint identifi_pk primary key (tb_cd_identifi));

create table tb_alumno
( tb_alumno_identifi int,
tb_alumno_password varchar(15),
tb_alumno_nombre varchar(50),
tb_alumno_direccio varchar(50),
tb_alumno_certific int,
constraint alumno_pk primary key (tb_alumno_identifi), constraint certific_fk
foreign key (tb_alumno_certific)
references tb_cd(tb_cd_identifi));

create table tb_materia
( tb_materia_identifi int,
tb_materia_nombre varchar(30),
constraint materia_pk primary key (tb_materia_identifi));

create table tb_notas
( tb_notas_materia int,
tb_notas_promedio float,
tb_notas_alumno int,
constraint alumno_fk foreign key (tb_notas_alumno) references
tb_alumno(tb_alumno_identifi));
```

CAPITULO 3

3.1 CONFIGURACION DE COMPONENTES

3.1.1 Instalación De White Box 3

- Seleccionar el lenguaje
- Disposición del teclado
- Configuración del mouse
- Pantalla de bienvenida
- Tipo de instalación (personalizada)
- Seleccionar la herramienta de particionar
- Definir las particiones
- Escoger la zona horaria
- Iniciar el formateo

3.2 CONFIGURACION DE LA RED

Observación:

Para realizar las siguientes configuraciones debe haber iniciado sesión como súper usuario (root) y disponer de un editor de texto que por defecto al momento de la instalación del sistema operativo es vi.

3.2.1 Configuración del Hostname

1. Editar el archivo **hosts** que se encuentra en el directorio de **etc**

   El comando a utilizar es el siguiente: **Vi /etc/hosts**

2. Luego se puede observar la información del archivo por default:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.0          localhost.localdomain localhost
```

3. Presione la tecla insert o la letra i para añadir la direccion ip, nombre y dominio del equipo. Debe quedar de la siguiente forma:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.0          localhost.localdomain   localhost
192.168.1.100      srvlinux.caos.com.ec    srvlinux
```

4. Se debe guardar los cambios, presionando la tecla **Escape** seguido del signo **:** (dos puntos) y las letras **wq .**

3.2.2Configuración del Sysconfig

1. Editar el archivo **network** que se encuentra en el directorio de **etc/sysconfig**

   El comando a utilizar es el siguiente:

   Vi/etc/sysconfig/network

2. Luego se puede observar la información del archivo por default:

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
```

3. Presione la tecla insert o la letra i para modificar el nombre del host y añadir la puerta de enlace Debe quedar de la siguiente forma:

   NETWORKING=yes
   HOSTNAME=srvlinux.caos.com.ec
   GATEWAY=192.168.1.100

Observacion : El Hostname del archivo **network** debe ser igual al que se encuentra en el archivo **hosts**

4. Se debe guardar los cambios, presionando la tecla **Escape** seguido del signo **:** (dos puntos) y las letras **wq** .

3.2.3 Configuración de las tarjetas de Red

1. Editar el archivo **ifcfg-eth0** que se encuentra en el directorio de **etc/sysconfig/network-scripts**

   El comando a utilizar es el siguiente: **Vi /etc/sysconfig/network- scripts/ifcfg-eth0**

2. Presione la tecla insert o la letra i para modificar el nombre del host y añadir la puerta de enlace Debe quedar de la siguiente forma:

   DEVICE=eth0
   ONBOOT=yes
   BOOTPROTO=static
   IPADDR=192.168.1.100
   NETMASK=255.255.255.0
   GATEWAY=192.168.1.100

3. Se debe guardar los cambios, presionando la tecla Escape seguido del signo **:** (dos puntos) y las letras **wq** .

3.3. Comprobar la Red

1. Comprobar si hay conectividad, ejecute el comando ping y la dirección ip que se quiere conectar, como se muestra a continuación.
   Ping 192.168.1.101

3.4. Configuración de los Servicios

Observación

- Para verificar si el paquete se encuentra instalado se utiliza el comando **rpm – q** y el nombre del paquete -.
  Ejemplo: **rpm –q Bind**
- Para instala el paquete se utiliza el comando rpm –ivh y el nombre del paquete
  Ejemplo: **rpm –ivh Bind**
- Para poder habilitar una línea se debe de eliminar el dnl al principio de la línea.
- Para desactivar una línea se debe de poner el dnl al principio de la línea que quiere deshabilitar.

3.4.1. Configuración del DNS

1. Para la configuración del DNS se deben tener levantados los siguientes paquetes de configuración:

   - ✓ Bind
   - ✓ Bind-libs
   - ✓ Bind-chroot
   - ✓ Bind-utils
   - ✓ Caching-nameserver

2. Editar el archivo **resolv.conf** que se encuentra en el directorio de **etc/resolv.con**

   El comando a utilizar es el siguiente: **Vi /etc/resolv.con**

3. Presione la tecla insert o la letra i para añadir el nombre del dominio , colocar la IP del equipo , debe quedar de la siguiente forma:

   Nameserver 192.168.1.100

4. Se debe guardar los cambios, presionando la tecla **Escape** seguido del signo **:** (dos puntos) y las letras **wq .**

5. Después de haber configurado los parámetros de las red deseados, se debe de reiniciar el servicio , ejecutando lo siguiente :

   Service network restart

6. Para poder Crear la Zona del dominio se debe de editar el archivo **caos.zone**

   El comando a utilizar es el siguiente **vi /var/named/chroot/var/named/caos.zone**

```
$TTL    86400
@       IN      SOA     dns.caos.       root.caos. (
                                2006031601      ; serial (d. adams)
                                28800           ; refresh
                                7200            ; retry
                                604800          ; expiry
                                86400           ; minimum
                                )
@       IN      NS      dns
@       IN      MX      10      mail
@       IN      A       192.168.1.100
srvlinux IN     A       192.168.1.100
srvweb  IN      A       192.168.1.101
pcwindows IN    A       192.168.1.23
www             IN      CNAME   srvlinux
mail            IN      A       192.168.1.100
dns             IN      CNAME   srvlinux
```

7. Para poder Crear la zona inversa se debe de ir a la siguiente ruta **vi /var/named/chroot/var/named/** y colar la ip de manera inversa sin el ultimo digito y colocar. **in-addr.arpa.zone**

   La IP es **192.168.1.100**

La forma inversa es **1.168.192.in-addr.arpa.zone**

/var/named/chroot/var/named/1.168.192.in-addr.arpa.zone

```
$TTL    86400
@       IN      SOA   dns.caos.     root.caos. (
                                2006031601   ; serial (d. adams)
                                28800        ; refresh
                                7200         ; retry
                                604800             ; expiry
                                86400        ; minimum
                                )
@       IN      NS    dns.caos.
1   IN      PTR   srvlinux.caos.
2   IN      PTR   srvweb.caos.
3   IN      PTR   pcwindows.caos.
```

```
zone "caos" IN {
    type master;
    file "caos.zone";
};

zone "1.168.192.in-addr.arpa.zone" IN {
    type master;
    file "1.168.192.in-addr.arpa.zone";
    allow-update        { none; };
};
```

8. Reiniciar el servicio : **service named restart**

3.4.1.1. Comprobar el DNS

Para comprobar que DNS fue levantado correctamente se digita el siguiente comando:

dig nombre_dominio

En nuestro caso seria: **dig caos**

Luego de este comando deberán aparecer las siguientes líneas que verifican el correcto funcionamiento:

```
; <<>> DiG 9.2.4 <<>> caos
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:
38263
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY:
1, ADDITIONAL: 0

;; QUESTION SECTION:
;caos.                   IN      A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
caos.           86400   IN      NS      dns.caos.

;; Query time: 83 msec
;; SERVER: 192.168.1.100 #53(192.168.1.100)
;; WHEN: Mon Marz 21 22:15:41 2007
;; MSG SIZE  rcvd: 61
```

3.4.2. Configuración de Sendmail

1. Para la configuración de SENDMAIL se deben tener levantados los siguientes paquetes de configuración:

   - ✓ Sendmail
   - ✓ Sendmail-cf
   - ✓ m4
   - ✓ Dovecot
   - ✓ Make
   - ✓ Cyrus –sasl
   - ✓ Cyrus –sasl-md5
   - ✓ Cyrus –sasl-plain

2. Para poder establecer dominios se debe de editar el archivo local-host-names que se encuentra en el directorio de **/etc/mail**

   El comando a utilizar es el siguiente: **Vi /etc/mail/local-host-names** , lo cual va aparecer lo siguiente :

   # local-host-names - include all aliases for your machine here.
   caos.com.ec

3. Para establecer dominios permitidos y poder enviar correo se debe de editar el archivo **relay-domains** que se encuentra en el directorio **/etc /mail**

   El comando a utilizar es el siguiente **vi /etc/mail/relay-domains**

4. Presione la tecla **insert** o la letra **i** para añadir el nombre del dominio que le van a dar permiso Debe quedar de la siguiente forma:

 # local-host-names - include all aliases for your machine here
caos.com.ec

Se debe guardar los cambios, presionando la tecla **Escape** seguido del signo : (dos puntos) y las letras **wq .**

5. Para definir la lista de control de acceso se debe de editar el archivo **access** que se encuentra en el directorio /etc/mail.

   El comando a utilizar es **vi/etc/mail/access** , se debe de añadir las direcciones ip a la cual le quiere dar acceso como se muestra a continuación .

```
# Check the /usr/share/doc/sendmail/README.cf file for a
description
# of the format of this file. (search for access_db in that file)
# The /usr/share/doc/sendmail/README.cf is part of the
sendmail-doc
# package.
#
# by default we allow relaying from localhost...
localhost.localdomain          RELAY
localhost                  RELAY
127.0.0.0                  RELAY
192.168.1.100              RELAY
192.168.1.101              RELAY
192.168.1.102                  RELAY
```

   Se debe guardar los cambios, presionando la tecla **Escape** seguido del signo **:** (dos puntos) y las letras **wq .**

6. Modificar el fichero **/etc/mail/sendmail.mc** y desactivar o habilitar funciones:

   Para poder recibir correo desde Internet o Lan localice la siguiente línea y eliminé Addr=127.0.0.1

```
DAEMON_OPTIONS(`Port=smtp, Addr=127.0.0.0, Name=MTA')dnl
DAEMON_OPTIONS(`Port=smtp, Name=MTA')dnl
```

- Para impedir que se acepte correo Spam se debe de comentar la siguiente línea

```
dnl FEATURE(`accept_unresolvable_domains')dnl
```

- Habilitar las siguientes líneas para definir la mascara que utilizara el servidor.

```
MASQUERADE_AS(`caos.com.ec')dnl
FEATURE(masquerade_envelope)dnl
FEATURE(masquerade_entire_domain)dnl
```

- Añadir al final el fichero lo siguiente el parámetro que defina el dominio.

  Cwcaos.com.ec

7. Para reiniciar servicio de **Sendmail** se debe de ejecutar lo siguiente :

   Service sendamail restart

3.4.3. Configuración del Dovecot

1. Editar el archivo **dovecot.conf** que se encuentra en el directorio de **etc**

   El comando a utilizar es el siguiente: **Vi /etc/dovecot.conf**
2. Presione la tecla **insert** o la letra **i** para añadir el imap pop3

   protocols = imap pop3
3. Se debe borrar el signo # para habilitar la siguiente línea y poner **yes**

   ssl_disable = yes
4. Se de borrar el signo # para habilitar la siguiente línea y agregar lo que se muestra después del signo **=** (igual)

   default_mail_env = mbox:~/mail/:INBOX=/var/mail/%u
5. Se de borrar el signo # para habilitar la siguiente línea y agregar lo que se muestra después del signo **=** (igual)

   mbox_locks = fcntl
6. Se debe guardar los cambios, presionando la tecla Escape seguido del signo **:** (dos puntos) y las letras **wq** .
7. Se debe de activar el servicio con el siguiente comando.

   Chkconfig imap on
   Chkconfig ipop3 on

3.4.5. Configuración del Apache

1. Para la configuración del HTTP se deben tener levantados el paquetes de configuración:
   - ✓ HTTPD
2. Se debe de añadir el servicio de arranque con el siguiente comando.

Chkconfig httpd on

3. Para iniciar el servicio por primera vez, solo basta utilizar:

   Service httpd Start

4. Para reiniciar el servicio, considerando que se interrumpirán todas las conexiones establecidas en ese momento, solo basta utilizar:

   Service httpd restart

3.4.4. Configuración del Openssl

1. Editar el archivo **openssl.cfn** que se encuentra en el directorio **usr/share/ssl**

El comando a utilizar es el siguiente: **Vi/usr/share/ssl/openssl.cfn**

2. Modificar los registró que se detallan:

```
dir             = /usr/local/caos
certs           = $dir/certs
crl_dir         = $dir/crl
database        = $dir/index.txt
new_certs_dir   = $dir/nuevoscerts
certificate     = $dir/certs/certsca.pem
serial          = $dir/serial
crl             = $dir/crl.pem
private_key     = $dir/privado/clavesca.pem
RANDFILE        = $dir/privado/.rand
x509_extensions     = usr_cert
```

3. Se debe de crear estructura a usar

```
cd /home/caos
mkdir ca
mkdir ca/certs
mkdir ca/crl
mkdir ca/nuevoscerts
mkdir ca/privado
mkdir ca/csr
echo "01" > ca/serial
touch ca/index.txt
```

CAPÍTULO 4

4.1 INSTALACION DE SOFTWARE REQUERIDO

4.1.1 Descargar JDK para Linux

Ir a la dirección: http://java.sun.com/javase/downloads/index.jsp

Hacer clik en la opción

JDK 5.0 Update 9

Hacer click en ACCEPT y luego escoger la opción

Linux Platform - J2SE(TM) Development Kit 5.0 Update 9		
Linux self-extracting file	jdk-1_5_0_09-linux-i586.bin	47.22 MB

Figura 4.1

4.1.2 Instalar JDK para Linux

1. Mediante una consola del sistema colóquese en el directorio donde desee realizar la instalación, se recomienda **/usr/local/**.
   cd /usr/local
2. Cambiar los permisos del archivo con el comando:
   chmod a+x j2sdk-1_4_2_<version>-linux-i586.bin
3. Ejecutar del directorio local :
   ./j2sdk-1_4_2_<numero_version>-linux-i586.bin
4. Aceptar la licencia de usuario, debe irse hasta el final del documento y digitar yes.
5. Una vez terminada la instalación se recomienda cambiar el nombre del directorio a **jdk;** quedando instalado en una ruta absoluta: **/usr/local/jdk .**

4.1.2.1. Configuración

El JDK requiere configurarse al igual que en los sistemas operativos Windows con diversas variables ambientales para su correcta operación.

Estas variables seran definidas a nivel global:

- ✓ **Nivel Global**: Permite que las variables sean accesibles para todo usuario del sistema; estas definiciones son colocadas en el archivo **/etc/profile** del sistema.

```
JAVA_HOME="/usr/local/jdk"
CLASSPATH="/usr/local/mislibrerias"
PATH="$PATH:/usr/local/jdk/bin"
export JAVA_HOME
export CLASSPATH
export PATH
```

4.1.2.2. Pruebas

1. Después de haber realizado la configuración de las variables de ambiente, es recomendable reiniciar el equipo para que realice los cambios. Una vez reiniciado, abra una consola del sistema y dentro de cualquier directorio invoque el siguiente comando:

   java -version

2. Si observa una respuesta indicando la versión del JDK lo ha instalado correctamente, en caso contrario realice los pasos anteriores hasta que esta prueba sea ejecutada correctamente.

```
root@srvlinux root]# java -version
java version "1.5.0_09"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_09-b03)
Java HotSpot(TM) Client VM (build 1.5.0_09-b03, mixed mode, sharing)
[root@srvlinux root]#
```

4.2. Instalación del Tomcat

1. Mediante una consola del sistema colóquese en el directorio donde desee realizar la instalación, se recomienda **/usr/local/**.

   cd /usr/local

2. Se debe de descomprimir el archivo con el siguiente comando:

   tar -xzvf jakarta-tomcat-5.x.x.tar.gz .

3. Una vez terminada la instalación se recomienda cambiar el nombre del directorio a **tomcat;** quedando instalado en una ruta absoluta:

   /**usr/local/tomcat.**

4.2.1. Configuración

El **Tomcat** requiere configurarse al igual que en los sistemas operativos Windows con diversas variables ambientales para su correcta operación.

Estas variables serán definidas a nivel global:

- ✓ **Nivel Global**: Permite que las variables sean accesibles para todo usuario del sistema; estas definiciones son colocadas en el archivo **/etc/profile** (a continuación de la configuración para java) del sistema.

```
CATALINA_HOME="/usr/local/tomcat"
PATH="$PATH:/usr/local/tomcat/bin"
export CATALINA_HOME
export PATH
```

4.2.2. Pruebas

1. Después de haber realizado la configuración de las variables de ambiente, es recomendable reiniciar el equipo para que realice los cambios. Una vez reiniciado, abra una consola del sistema y dentro de cualquier directorio invoque el siguiente comando:

   /usr/local/tomcat5.5.11/bin/startup.sh

2. Se asignara las variables de entorno dando como resultado lo siguiente :

   Using CATALINA_BASE: /usr/local/tomcat5.5.11
   Using CATALINA_HOME: /usr/local/tomcat5.5.11
   Using CATALINA_TMPDIR: /usr/local/tomcat5.5.11/temp
   Using JRE_HOME: /usr/local/jdk1.5/jre

3. Luego se va abrir el navegador con la dirección IP asignada del servidor por el puerto **8080**

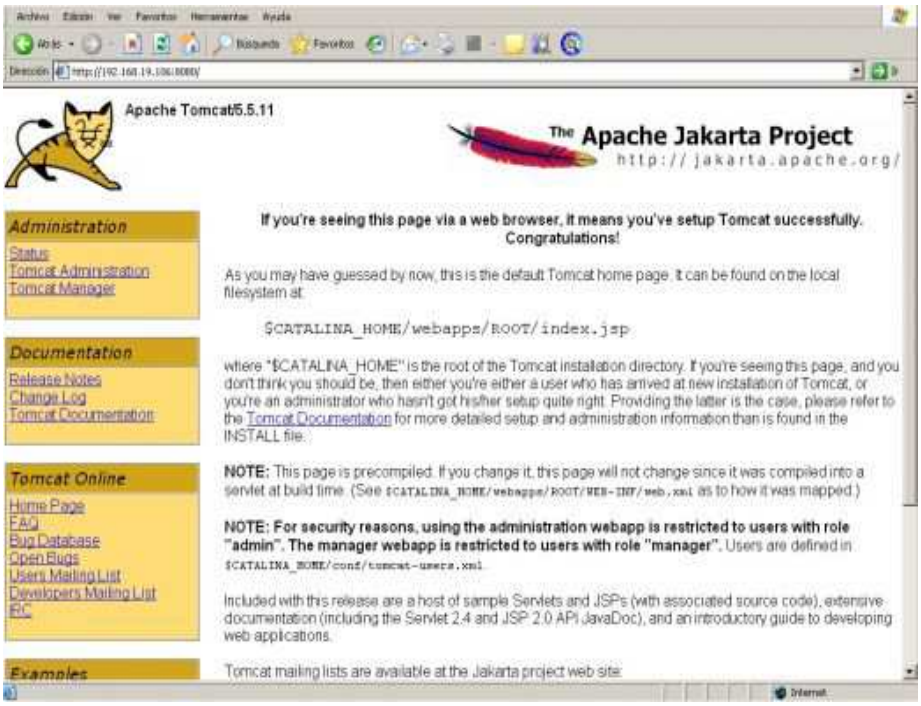
Apache Tomcat/5.5.11
The Apache Jakarta Project
http://jakarta.apache.org/
Administration
Status
Tomcat Administration
Tomcat Manager
Documentation
Release Notes
Change Log
Tomcat Documentation
Tomcat Online
Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC
Examples
If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!
As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:
$CATALINA_HOME/webapps/ROOT/index.jsp
where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.
NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See $CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml as to how it was mapped.)
NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in $CATALINA_HOME/conf/tomcat-users.xml.
Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.
Tomcat mailing lists are available at the Jakarta project web site:

Figura 4.2

4.3. Instalación de MySql

1. Colóquese en el directorio donde tenga alojado Mysql y digite lo siguiente :

   Rpm – ivh l**Mysql** dar enter y automáticamente se instala

2. Mysql se guarda en el directorio **var/lib/mysql** donde tendrá alojada toda la base de datos.

CAPÍTULO 5

5.1 CONFIGURACION DEL SISTEMA CAOS

5.1.1. Cofiguracion Del Keystore

1. Mediante una consola del sistema colóquese en el directorio **/urs/local/jdk** en el cual se debe de digitar lo siguiente :

keytool -genkey -alias tomcat -keyalg RSA

2. Luego en la pantalla solicita que ingrese lo siguiente datos

Escriba la contraseña del almacen de claves:
┐Cuales son su nombre y su apellido?
[Unknown]: CAOS
┐Cual es el nombre de su unidad de organizacion?
[Unknown]: CAOS
┐Cual es el nombre de su ciudad o localidad?
[Unknown]: guayaquil
┐Cußl es el nombre de su estado o provincia?
[Unknown]: guayas
┐Cußl es el codigo de pais de dos letras de la unidad?
[Unknown]: EC
┐Es correcto CN=CAOS, OU=CAOS, O=CAOS, L=guayaquil, ST=guayas, C=EC?

5.2. Configuración del Tomcat

1. Mediante una consola del sistema colóquese en el directorio **/usr/local/tomcat./conf**
2. Se editar el archivo **vi /usr/local/tomcat./conf/server.xml** y se debe de habilitar las siguientes líneas.

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->

  <Connector port="8443"
        maxThreads="150"                    minSpareThreads="25"
maxSpareThreads="75"
        enableLookups="false" disableUploadTimeout="true"
        acceptCount="100" debug="0" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS"
            keystorePass="  "
            keystoreFile="usr\local\java\.keystore"
/>
```

5.3. Configurar Eclipse

1. El proyecto se debe de copiar en la ruta **/usr/local/eclipse/workspace**
2. Se abre Eclipse y se debe de escoger las siguientes Opciones
   - ✓ File
   - ✓ Import escoja la opción proyecto existente en workspace.
   - ✓ Busca donde se encuentra ubicada el proyecto , siguiente

3. Configurar Tomcat en Eclipse y se debe de seguir los siguientes pasos.
   - ✓ Dar clic en el icono de BD
   - ✓ Escoger la configuere Server
   - ✓ Escoger Tomcat 5 y ubicar la rutas **/usr/local/apache /tomcat 5.0**, OK
   - ✓ En la opción de tomcat Server ponerle Enable

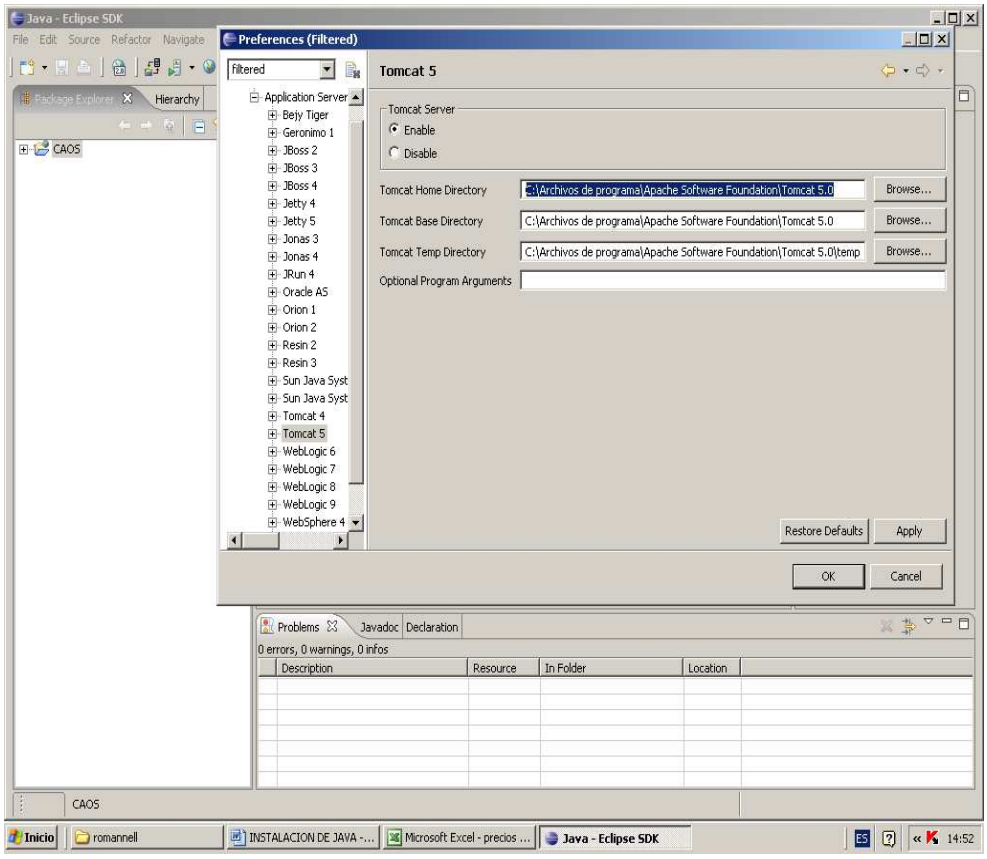

FIGURA 5.1

4. Configurar Java en Eclipse y se debe de seguir los siguientes pasos.
   - ✓ Dar clic en el icono de BD
   - ✓ Escoger la configuere Server
   - ✓ Escoger Tomcat 5
   - ✓ Escoger JDK , dar clip en Add y buscar la ruta donde se encuentra Java **/usr/local/java/jdk** , y dar clic en OK

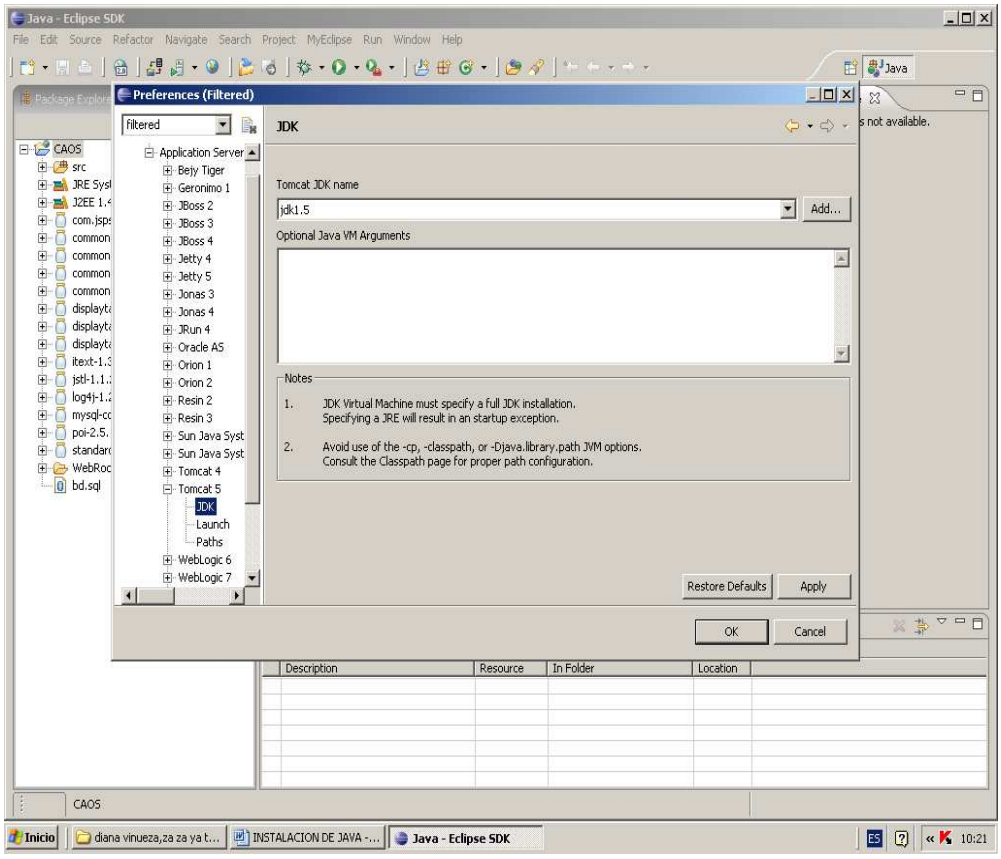

FIGURA 5.2

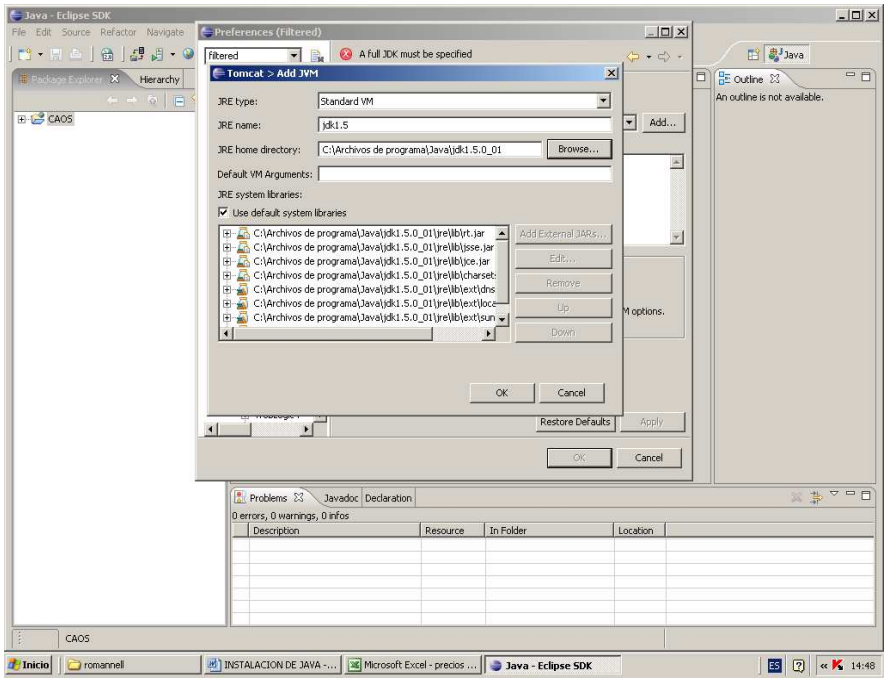

FIGURA 5.3

5. Hacer un Deployments debe de seguir los siguientes pasos

- ✓ Clip derecho a la carpeta de CAOS y se escoje Myclipse , add remove and Deployments Se agrega el tomcat 5
- ✓ Packagegd Archive y finish

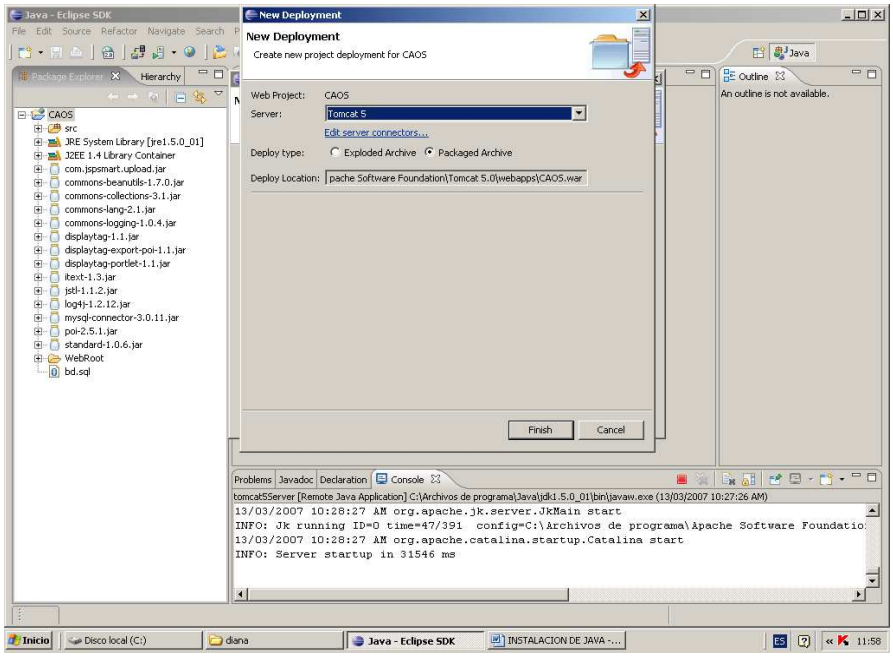

FIGURA 5.4

CAPITULO 6

6.1 CODIGO

En este capitulo se detallara el código mas relevante del proyecto.

6.1 Código de conexión

```
package beans;
import java.sql.*;
public class Conexion
{
        public static Connection getConexion()
        {
            String url="jdbc:mysql://"+Parametros.host_bd+"/"+Parametros.bd;
           Connection conexion=null;
           try
            {
               Class.forName("com.mysql.jdbc.Driver");
               conexion =
  DriverManager.getConnection(url,Parametros.usuario_bd,Parametros.clave_bd);
                                        }
            catch(Throwable e)
            {      e.printStackTrace();          }
            return conexion;
        }

        public static int insertaActualiza(String sql)
        {
                 int resultado=-1;
                Connection conexion =getConexion();
                if(conexion!=null)
                {
                     Try
                     {
                                Statement st = conexion.createStatement();
                                resultado = st.executeUpdate(sql);
                                st.close();
                                conexion.close();
                      } catch(Throwable exc)
                        {
                                exc.printStackTrace();
                                resultado=-1;
                         }
```

```
                    }
                     return resultado;
                }
 public static String maxCampoTabla(String campo,String tabla)
 {
       String resultado=null;
       String sql="select max("+campo+") from "+tabla;
       Connection conexion =Conexion.getConexion();
        if(conexion!=null)
       {
            try
              {
                  Statement st = conexion.createStatement();
                  ResultSet rs = st.executeQuery(sql);
                  if(rs.next())
                  {       resultado=rs.getString(1);          }
                   rs.close();
                   st.close();
                   conexion.close();
               }  catch(Throwable exc)
                     {     exc.printStackTrace();    }
        }
         return resultado;
    }
}
```

6.2. Codigo para encriptar y desencriptar Clave

```
package beans;
import java.io.UnsupportedEncodingException;
import java.security.InvalidAlgorithmParameterException;
import java.security.spec.AlgorithmParameterSpec;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.KeySpec;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.PBEParameterSpec;

public class Cifrador
{
  private static byte[] SALT_BYTES = {  (byte)0xA9, (byte)0x9B, (byte)0xC8,
  (byte)0x32,  (byte)0x56, (byte)0x35, (byte)0xE3, (byte)0x03      };

  private static int ITERATION_COUNT = 19;

  public static String encriptar(String passPhrase, String str)
```

```
        {
            Cipher ecipher = null;
            Cipher dcipher = null;
            try
            {
                // Crear la key
                KeySpec keySpec = new PBEKeySpec(passPhrase.toCharArray(),
                                            SALT_BYTES, ITERATION_COUNT);
                SecretKey key = SecretKeyFactory.getInstance
                                    ("PBEWithMD5AndDES").generateSecret(keySpec);
                ecipher = Cipher.getInstance(key.getAlgorithm());
                dcipher = Cipher.getInstance(key.getAlgorithm());
                // Preparar los parametros para los ciphers
                AlgorithmParameterSpec paramSpec  =  new PBEParameterSpec(
                                                    SALT_BYTES, ITERATION_COUNT);
                // Crear los ciphers
                 ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
                dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
              }   catch (javax.crypto.NoSuchPaddingException e)
                  {    e.printStackTrace();  }
                  catch (java.security.NoSuchAlgorithmException e)
                  {   e.printStackTrace();   }
                  catch (java.security.InvalidKeyException e)
                  {   e.printStackTrace();   }
                  catch (InvalidKeySpecException e)
                  {    e.printStackTrace();   }
                  catch (InvalidAlgorithmParameterException e)
                  {    e.printStackTrace();   }
         try
         {
             // Encodear la cadena a bytes usando utf-8
               byte[] utf8 = str.getBytes("UTF8");
             // Encriptar
                byte[] enc = ecipher.doFinal(utf8);
             // Encodear bytes a base64 para obtener cadena
                return new sun.misc.BASE64Encoder().encode(enc);
          }
              catch (javax.crypto.BadPaddingException e)
              {                    }
             catch (IllegalBlockSizeException e)
              {   }
             catch (UnsupportedEncodingException e)
              {   }
          return null;
}
```

```
public static String desencriptar(String passPhrase, String str)
{
    Cipher ecipher = null;
    Cipher dcipher = null;
    try
    {    // Crear la key
         KeySpec keySpec = new PBEKeySpec(passPhrase.toCharArray(),
                                  SALT_BYTES, ITERATION_COUNT);
         SecretKey key = SecretKeyFactory.getInstance(     "PBEWithMD5AndDES").
                                 generateSecret(keySpec);
         ecipher = Cipher.getInstance(key.getAlgorithm());
         dcipher = Cipher.getInstance(key.getAlgorithm());
        // Preparar los parametros para los ciphers
        AlgorithmParameterSpec paramSpec = new PBEParameterSpec(SALT_BYTES,
                                                    ITERATION_COUNT);
         // Crear los ciphers
          ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
          dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
    }
       catch (javax.crypto.NoSuchPaddingException e)
       {            e.printStackTrace();          }
       catch (java.security.NoSuchAlgorithmException e)
       {           e.printStackTrace();          }
       catch (java.security.InvalidKeyException e)
       {           e.printStackTrace();          }
       catch (InvalidKeySpecException e)
       {           e.printStackTrace();          }
       catch (InvalidAlgorithmParameterException e)
       {           e.printStackTrace();          }
     try
     {
             // Decodear base64 y obtener bytes
                byte[] dec = new sun.misc.BASE64Decoder().decodeBuffer(str);
             // Desencriptar
                byte[] utf8 = dcipher.doFinal(dec);
             // Decodear usando utf-8
               return new String(utf8, "UTF8");
      }   catch (javax.crypto.BadPaddingException e)
          {          e.printStackTrace();          }
          catch (IllegalBlockSizeException e)
          {          e.printStackTrace();          }
          catch (UnsupportedEncodingException e)
          {          e.printStackTrace();          }
          catch (java.io.IOException e)
          {         e.printStackTrace();          }
          return null;
  }
}
```

6.3. Código para crear el Certificado

```
package beans;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.io.*;
public class Certificado
{
    public Certificado()
    {       // TODO Apéndice de constructor generado automáticamente      }

    public static boolean publicarenCRL(String cod_solicitud,String cod_usuario)
    {
        boolean resultado=false;
        String sql="select b.t_cd_numero from t_solicitud a,t_cd b  where
        a.t_solicitudrv_cd=b.t_cd_numero and a.t_solicitudrv_numero="+
                                                              cod_solicitud;
       Connection conexion =Conexion.getConexion();
        if(conexion!=null)
        {  try
                {
                   Statement st = conexion.createStatement()
                   ResultSet rs = st.executeQuery(sql);
                   if(rs.next())
                   {
                       String codigo_cd=rs.getString(1);
                       SimpleDateFormat formato = new SimpleDateFormat("yyyy-
                                                                    MM-dd");
                       String fecha_revocacion = formato.format(new Date());
                       sql="update t_cd set t_cd_publicado=1 where t_cd_numero="
                                                             +codigo_cd;
                       if(Conexion.insertaActualiza(sql)>=0)
                       sql="insert into t_crl(t_crl_CD,t_crl_fechaRV,t_crl_username)
                     values("+codigo_cd+",'"+fecha_revocacion+"',"+cod_usuario+")";

                      if(Conexion.insertaActualiza(sql)>=0)
                       {       resultado=true;          }
                     }
                   }
                   rs.close();
                   st.close();
                   conexion.close();
                   } catch(Throwable exc)
                     {       exc.printStackTrace(); }
                }
```

```
                return resultado;
               }
public static boolean crearCertificado(String cod_solicitud,String identidad)
{
    try
   {
         String comando_crear=Parametros.generador+
         " -genkey -alias "+cod_solicitud+
          -keypass "+Parametros.keypassdefault+cod_solicitud+
          " -keyalg "+Parametros.keyalg+
          " -keysize "+Parametros.keysize+
          " -validity "+Parametros.validez+
          " -keystore "+Parametros.keystore+
          " -storepass "+Parametros.storepass+
          " -dname "+identidad;
          Ejecuta_programs.ExecuteCommand(comando_crear);
     }   catch (IOException e)
         {        e.printStackTrace();
                 return false;
         }
       return true;
}

 public static boolean exportarCertificado(String cod_solicitud)
{
     try{
            String comando_crear=null;
            comando_crear=Parametros.generador+
            " -export -alias "+cod_solicitud+
            " -keypass "+Parametros.keypassdefault+cod_solicitud+
            " -file "+cod_solicitud+".cer"+
            " -keystore "+Parametros.keystore+
            " -storepass "+Parametros.storepass;

           Ejecuta_programs.ExecuteCommand(comando_crear);
         }      catch (IOException e)
                {            return false;       }
        return true;
}

public static boolean creaCertificado()
{    return true;    }

public static boolean existeCertificadoActivo(String cod_usuario)
{
      boolean resultado=false;
      String sql="select count(a.t_solicitudrv_cd) from t_solicitud a, t_cd b where
          a.t_solicitudrv_tipo=1 and a.t_solicitudrv_estado=6 and (a.t_solicitudrv_cd is
          not null) and a.t_solicitudrv_cd=b.t_cd_numero and b.t_cd_status=1 and
          a.t_solicitudrv_registro='"+cod_usuario+"'";
```

```
Connection conexion =Conexion.getConexion();

 if(conexion!=null)
{    try {
              Statement st = conexion.createStatement();
              ResultSet rs = st.executeQuery(sql);
               if(rs.next())
              {
                    if(rs.getInt(1)>0)
                     {          resultado=true;    }
               }
               rs.close();
               st.close();
               conexion.close();
            }  catch(Throwable exc)
               {      exc.printStackTrace(); }
             }
         return resultado;
    }


   private static String completa(String frase,int longitud,String caracter)
   {
        String resultado=frase;
        while(resultado.length()<longitud)
        {       resultado=caracter+resultado;       }
        return resultado;
     }


  public static String getCertificados(String cod_usuario)
 {
        String resultado="";
        String sql="select a.t_solicitudrv_numero from t_solicitud a, t_cd b where
                        a.t_solicitudrv_tipo=1 and a.t_solicitudrv_estado=6 and
                       (a.t_solicitudrv_cd is not null) and a.t_solicitudrv_cd=
                         b.t_cd_numero and b.t_cd_status=1 and
                         t_solicitudrv_registro= '" +cod_usuario+"'
                         order by a.t_solicitudrv_numero";
         Connection conexion =Conexion.getConexion();
         if(conexion!=null)
         {
              try
              {
                  Statement st = conexion.createStatement();
                  ResultSet rs = st.executeQuery(sql);
                  while(rs.next())
                  {
                      String numero_solicitud=rs.getString(1);
                      resultado=resultado+"<option value='"+numero_solicitud+"' />"
```

```
                                    +numero_solicitud;
                        }
                        rs.close();
                        st.close();
                        conexion.close();
                    }   catch(Throwable exc)
                        {           exc.printStackTrace(); }
                }
            return resultado;
        }


public static String getEstadoSolicitudes(String cod_usuario)
{
    String resultado="";
    String sql="select a.t_solicitudrv_numero,b.t_estadoSolicitud_descripcion from
                t_solicitud a, t_estadosolicitud b where a.t_solicitudrv_tipo=1 and
                a.t_solicitudrv_estado=b.t_estadoSolicitud_codigo and
                a.t_solicitudrv_registro='"+cod_usuario+"' order by
                a.t_solicitudrv_numero";
     Connection conexion =Conexion.getConexion();
     if(conexion!=null)
     {  try  {
                Statement st = conexion.createStatement();
                ResultSet rs = st.executeQuery(sql);
               while(rs.next())
               {
                   String numero_solicitud=rs.getString(1);
                  String estado_solicitud=rs.getString(2);
                  String value_solicitud=completa(numero_solicitud,5,"0");

                   value_solicitud=value_solicitud+estado_solicitud;
                   resultado=resultado+"<option value='"+value_solicitud+"' />"
                            +numero_solicitud;
                }
               rs.close();
               st.close();
               conexion.close();
           } catch(Throwable exc)
             {   exc.printStackTrace();                  }
        }
       return resultado;
}

public static boolean validaRevocatoria(String solicitado_por,String numero_solicitud)
{
    boolean resultado=false;
    String sql="select a.t_entidad_nombreresponsable,a.t_entidad_apellidoresponsable
                from t_entidad a, t_solicitud b where a.t_entidad_codigo=
                b.t_solicitudrv_entidad and b.t_solicitudrv_numero="
```

```
                +numero_solicitud;
    Connection conexion =Conexion.getConexion();
            if(conexion!=null)
            {  try  {
                        Statement st = conexion.createStatement();
                        ResultSet rs = st.executeQuery(sql);
                        if(rs.next())
                        {
                            String nombre=rs.getString(1);
                            String apellido=rs.getString(2);
                            String cadena_mix=nombre+" "+apellido;

                             if(cadena_mix.compareTo(solicitado_por)==0)
                            {               resultado=true;          }
                         }
                                rs.close();
                                st.close();
                                conexion.close();
                      }  catch(Throwable exc)
                         {        exc.printStackTrace();          }
                }return resultado;
     }

public static boolean revocarCertificado(String numero_solicitud,String solicitado_por,
                                        String cod_motivo)
{
        SimpleDateFormat formato = new SimpleDateFormat("yyyy-MM-dd");
        String fecha_creacion = formato.format(new Date());                String
        sql="update t_solicitud set
        t_solicitudrv_estado=1,t_solicitudrv_tipo=2,t_solicitudrv_fechacrea='"+fecha_cr
        eacion+"',t_solicitudrv_motivo="+cod_motivo+",t_solicitudrv_hechopor='"+soli
        citado_por+"' where t_solicitudrv_numero="+numero_solicitud;
        if(Conexion.insertaActualiza(sql)>0)
         {       return true;               }
        return false;
}
}
```

MANUAL DE USUARIO

INTRODUCCIÓN

Este documento es una ayuda para todo aquel usuario del Sistema ***AUTORIDAD CERTIFICADORA OPEN SOURCE (CAOS), PAGINA WEB DE CONSULTA DE NOTAS*** Léalo detenidamente antes de empezar el uso de la aplicación.

Para evitar inconvenientes debido al no conocimiento de su correcto funcionamiento y uso, todo usuario final del sistema debe basar su criterio en este documento.

CAPITULO 1

1.1 USO DEL SISTEMA CAOS

- Para ingresar a la página de la Unidad Certificadora CAOS se abre un navegador y se coloca **HTTPS://LOCALHOST/CAOS** O la IP que se encuentra ubicada el **CAOS**, con ello se logra ejecutar el sistema y se tendrá la primera pantalla.(Ver figura 1)

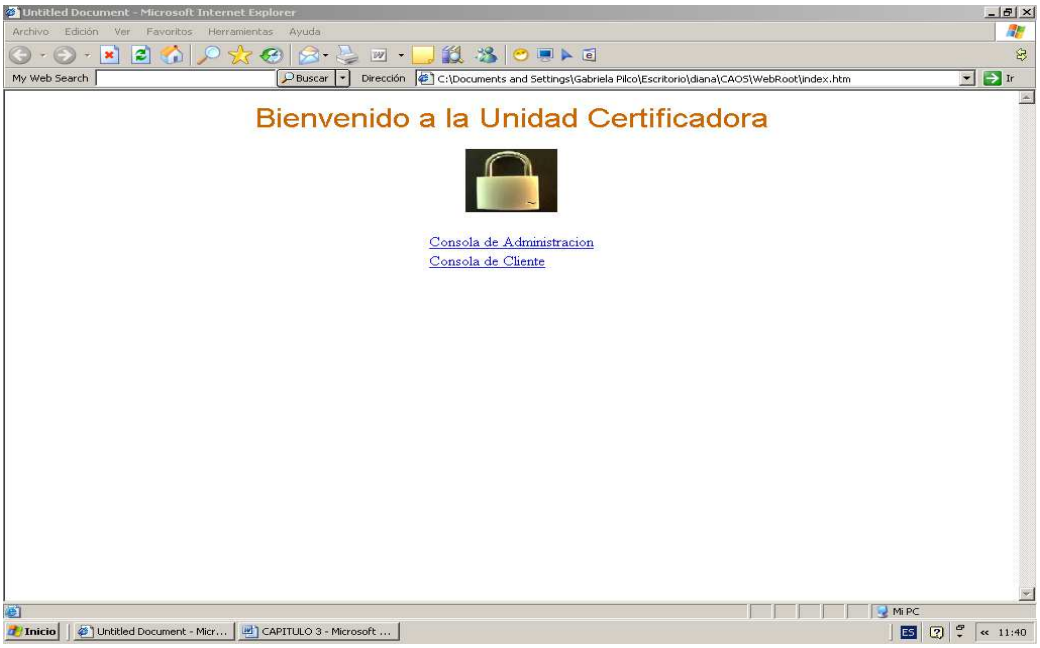

FIGURA 1.1

- En la consola de Administración solo va a tener acceso el Administrador y se tendrá la siguiente pantalla.(Ver figura 1.2)

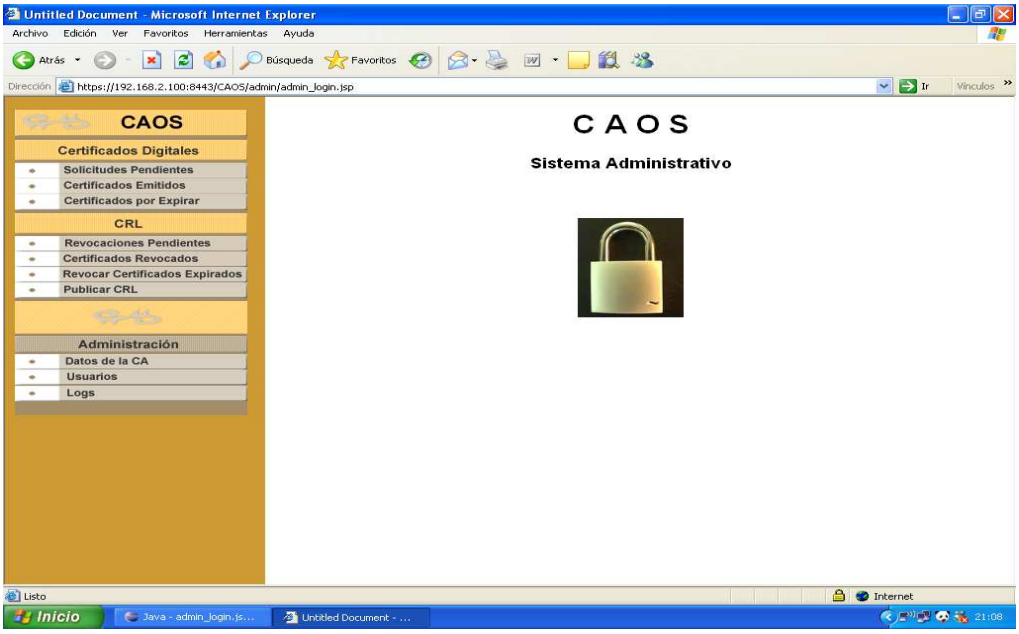

FIGURA 1.2

- En la consola del cliente primero deberá registrarse dar click en Registro se tendrá la siguiente pantalla.(Ver figura 1.3 , figura 1.4)

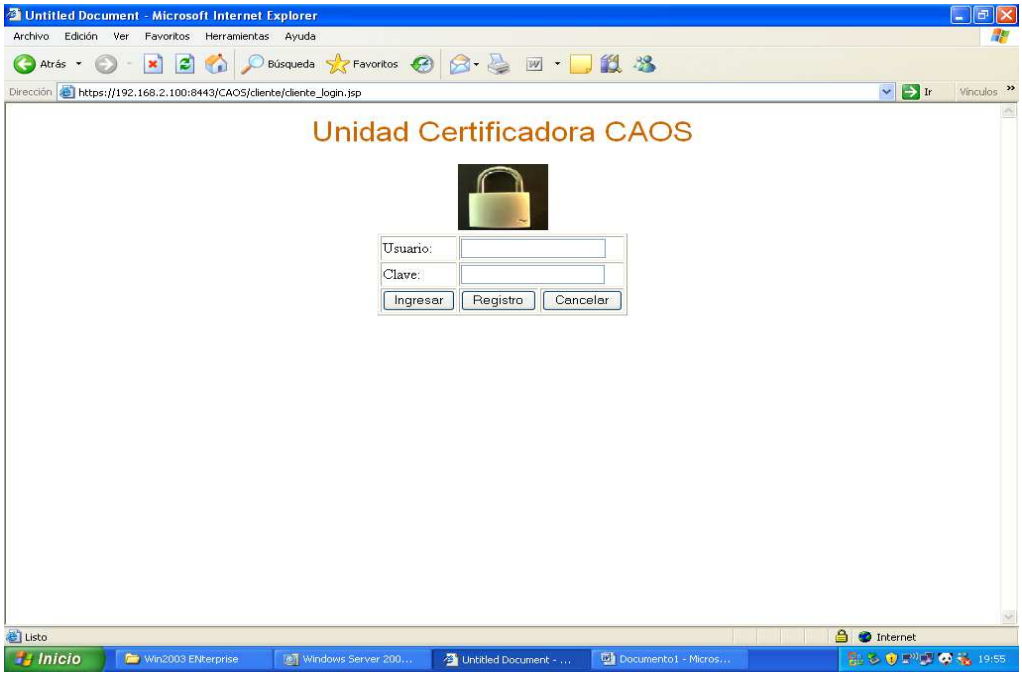

FIGURA 1.3

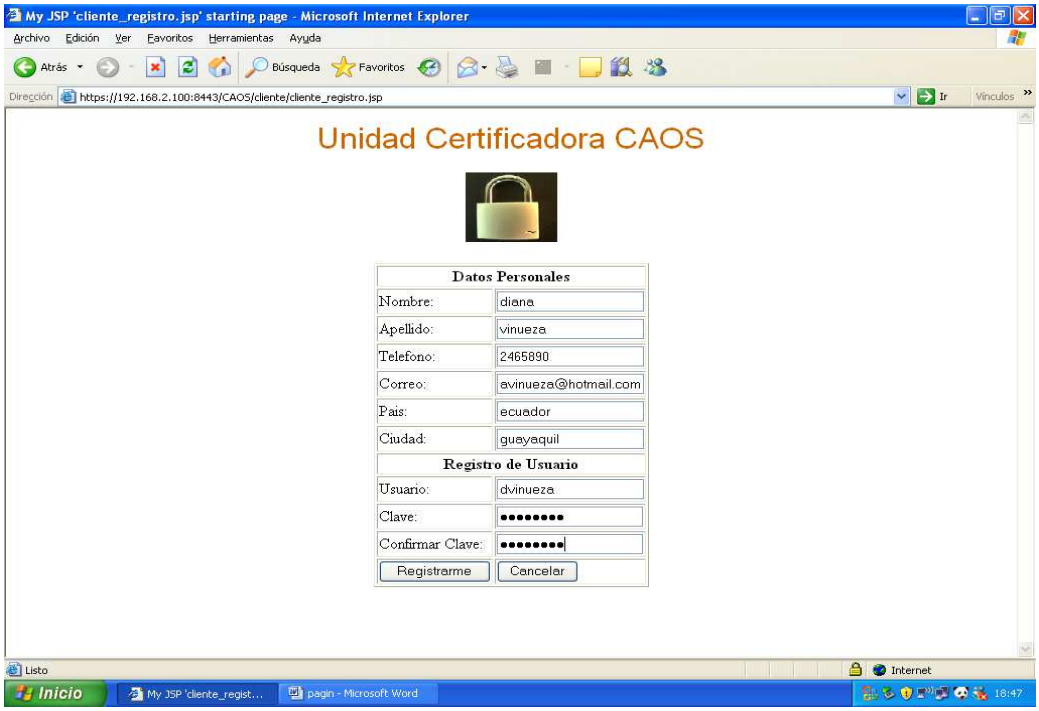

FIGURA 1.4

- Una vez que se registraron , ingresar el usuario y contraseña dar click en la opción de **ingresar** se tendrá la siguiente pantalla.(Ver figura 1.5 , figura 1.6)

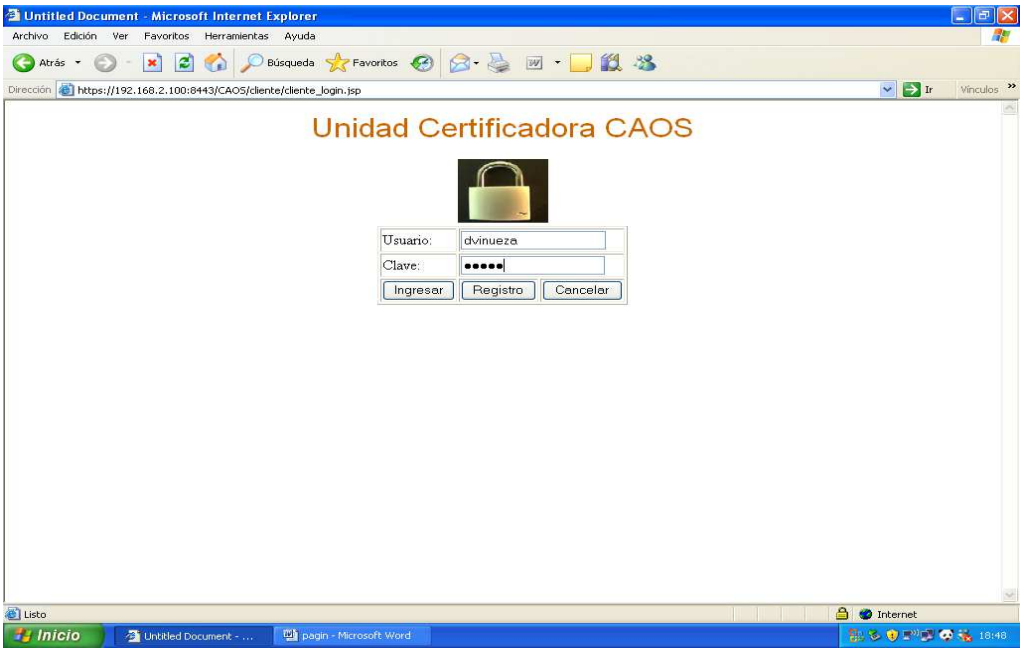

FIGURA 1.5

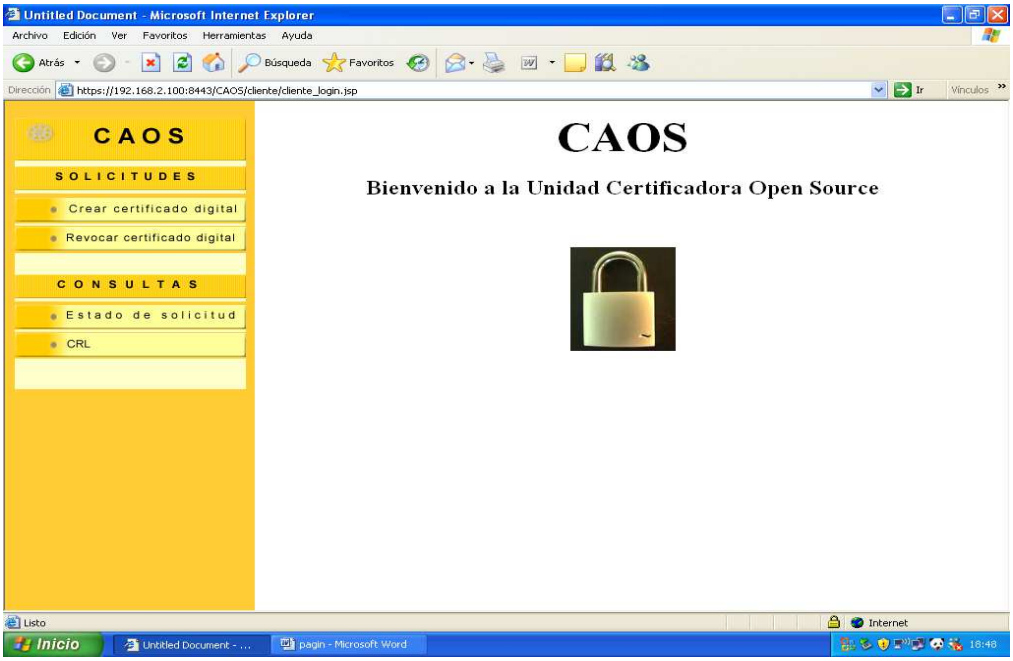

FIGURA 1.6

- Para que el cliente solicite un certificado, el cliente deberá escoger la opción que se encuentra en la parte izquierda crear certificado digital dar clic y le aparecerá en la parte derecha el tipo de entidad el deberá escoger si es Personal o Servidor dependiendo de la necesidad del cliente., se tendrá la siguiente pantalla.(Ver figura 1.7)

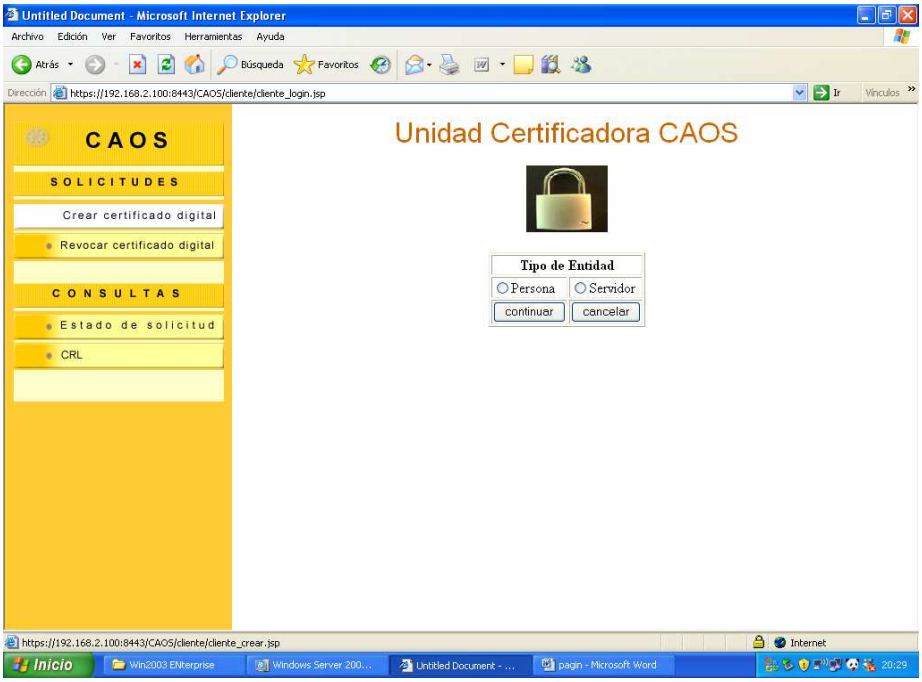

FIGURA 1.7

- si selecciona la opción de Persona se tendrá la siguiente pantalla.(Ver figura 1.8)

  Nota : la fecha deberá colocarse con el siguiente formato : Año- Mes- Dia

FIGURA 1.8

- Si selecciona la opción de Servidor se tendrá la siguiente pantalla.(Ver figura 1.9)
- Nota : la fecha deberá colocarse en el siguiente orden : Año/Mes/Dia

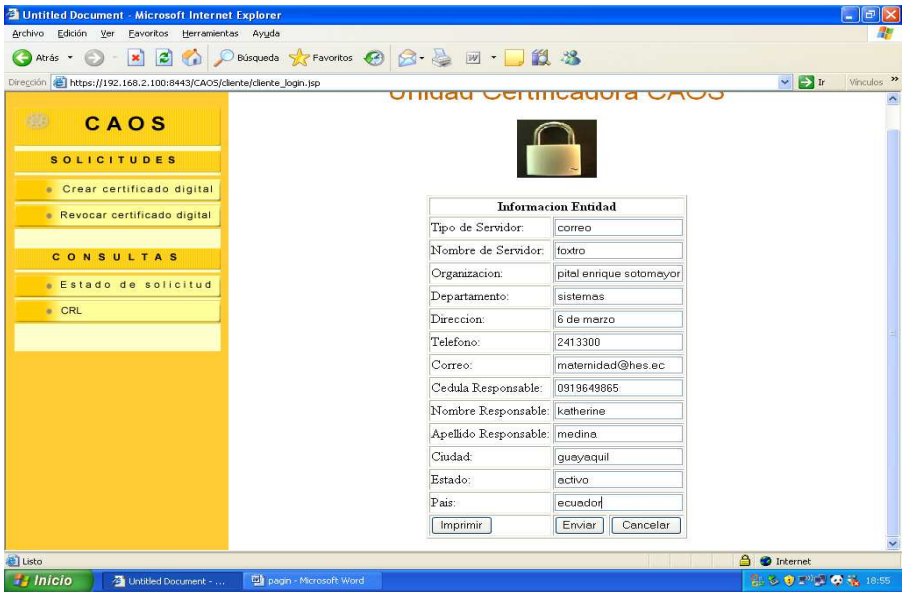

FIGURA 1.9

- En la opción Estado de Solicitud se mostrara el estado que se encuentra los certificados si es nuevo si esta en espera etc tendrá la siguiente pantalla.(Ver figura 1.10)
- En la opción de CRL los clientes podrán revocar los certificados donde llenara la solicitud indicando el motivo el cual lo quiere revocar

1.2. USO DE LA PÁGINA WEB DE CONSULTA DE NOTAS

- Para ingresar a la página Web de Consulta de Notas se abre un navegador y se coloca **HTTP://192.168.1.101/academico**, con ello se logra ejecutar el sistema y se tendrá la primera pantalla.(Ver figura 2)

FIGURA 2.1

FIGURA 2.2

- Una vez situado en la página principal de la Web, se deberá digitar el Usuario y password suministrado por la universidad, para acceder al menú principal de Información. y se tendrá la siguiente pantalla.(Ver figura 2.1)

FIGURA 2.3

Información.

Dentro de esta opción podremos encontrar información el proyecto de tesis “Consulta de Notas” desarrollado, tales como recursos utilizados, esquema del proyecto y más.

También tendremos Un menú para la facilidad en el acceso a las demás opciones de la Web

Consulta de notas

El ingreso a la esta opción se realizará mediante la utilización de un certificado digital, suministrado y validado por la C.A. (Autoridad certificadora) creada para brindar seguridad en el acceso a esta información.

Los pasos para la solicitud y obtención de este certificado se detallan en el manual de Usuario de la C.A.

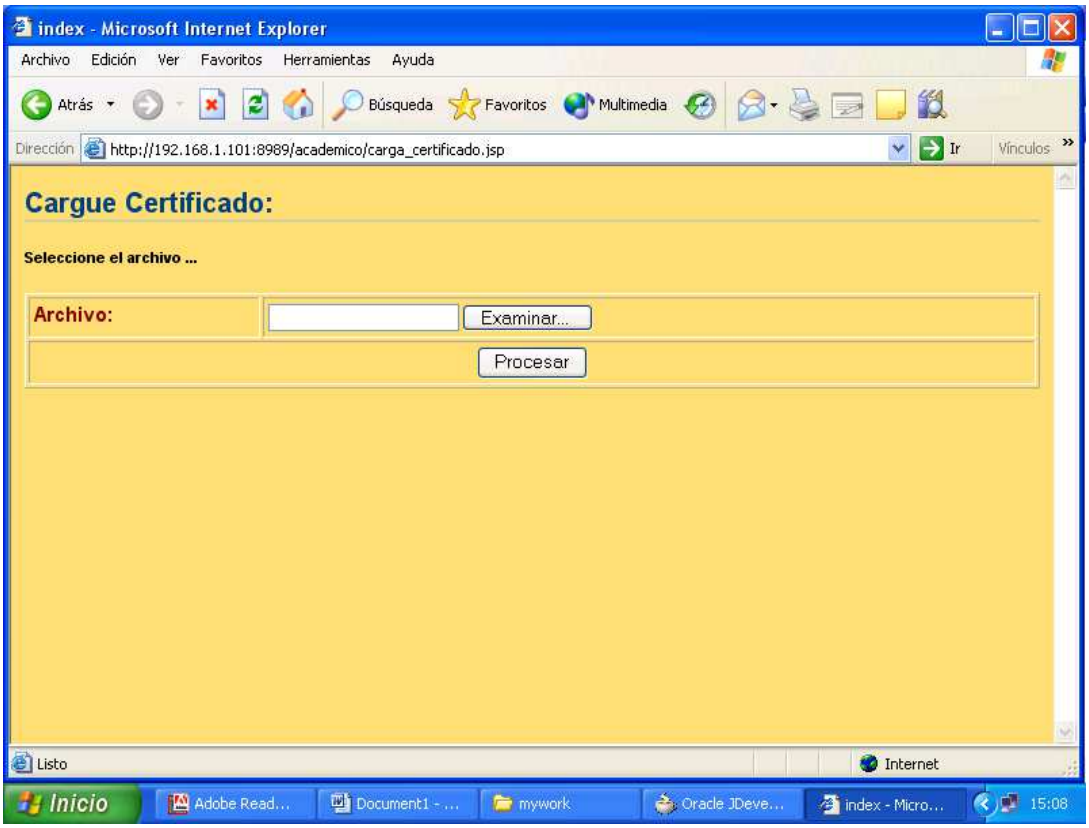

FIGURA 2.4

- Una vez que se haya verificado la validez del certificado se abrirá una página con las notas correspondientes al usuario. y se tendrá la siguiente pantalla.(Ver figura 2.3)

FIGURA 2.5

Información de la Carrera

Si se da clic en esta opción se podrá consultar información general sobre la carrera de Ingeniería en Sistemas

Cambio de password

- La pagina Web posee la opción de cambio de password con la finalidad de brindar un grado mayor de seguridad a los usuarios que realizarán la consulta de notas y se tendrá la siguiente pantalla.(Ver figura 2.4)

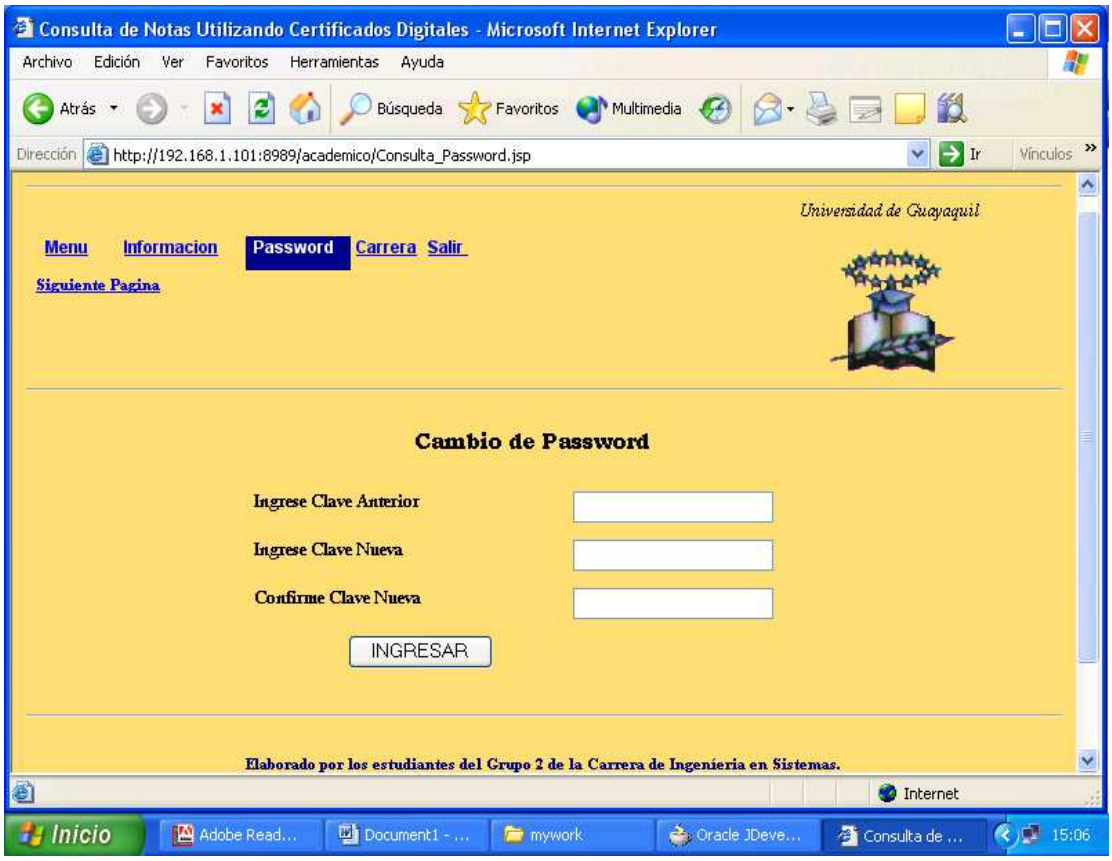

FIGURA 2.6

Manual de Técnico

INDICE DE FIGURAS

	Pag
Figura1.1 Representación del diagrama de Clase	2
Figura 1.2 Diagrama de clases para la Unidad Certificadora	2
Figura 1.3 Diagrama de clases para la página Web de Consulta de Notas	3
Figura 1.4 Estándar de la base de datos para la Unidad Certificadora.	3
Figura 1.5 Ejemplo de la Tablas	4
Figura 1.6 Estándar de la base de datos para la Consulta de Notas.	4
Figura 1.7 Ejemplo de la Tablas	4
Figura 1.8 Atributos para la Unidad Certificadora	6
Figura 1.9 Atributos para la Consulta de Notas	6
Figura 1.10 DER para la Unidad Certificadora	7
Figura 1.11 DER para la Consulta de Notas	8
Figura 2.1 Diccionario de datos para Unidad Certificadora	9
Figura 2.2 Diccionario de datos para consulta de Notas	9
Figura 4.1 Descargar JDK para Linux	29
Figura 5.1 Configurar Tomcat	31
Figura 5.5 Configurar Java en Eclipse	32
Figura 5.3 Opción de Java en Eclipse	33
Figura 5.4 Deployments	33

Manual de usuario

INDICE DE FIGURAS

Pag

Figura 1.1 Ingreso a la pagina de CAOS 45
Figura 1.2 Consola del Administración 46
Figura 1.3 Consola del Cliente 46
Figura 1.4 Consola del Cliente llenar datos 47
Figura 1.5 Consola de Login 47
Figura 1.6 Pantalla de Bienvenida 48
Figura 1.7 Pantalla de Solicitud del certificado 48
Figura 1.8 Pantalla de solicitud para la Persona 49
Figura 1.9 Pantalla de solicitud para el servidor 49
Figura 2.1 Pantalla de bienvenida de Consulta de Notas 50
Figura 2.2 Pantalla de Login de Consulta de Notas 51
Figura 2.3 Pantalla de menú de Consulta de Nota 51
Figura 2.4 Pantalla de petición del Certificado 52
Figura 2.5 Pantalla de Notas 53
Figura 2.6 Pantalla de Cambio de Password 54

TABLA DE CONTENIDO TOMO II

MANUAL TECNICO

INTRODUCCIÓN ***1***

CAPITULO 1 ***2***

1.1 DIAGRAMA DE CLASES **2**
1.1.1 Diagrama de clases para la Unidad Certificadora. 2
1.1.2 Diagrama de clases para la Consulta de notas 3

1.2 DISEÑO DE LA BASE DE DATOS **3**
1.2.1 Estándares de La Base de Datos 3
1.2.1.1 Estándar de la base de datos para la Unidad Certificadora. 3
1.2.1.2 Estándar de la base de datos para la consulta de notas. 4
1.2.2 Definición de Entidades 5
1.2.3 Atributos por Entidad 6
1.2.4 Atributos para la Unidad Certificadora 6
1.2.5 Atributos para la consulta de notas. 6

1.3 Diagrama Entidad Relación **7**
1.3.1 DER para la Unidad Certificadora 7
1.3.2 DER para la consulta de notas 8

CAPÍTULO 2 ***9***

2.1 DICCIONARIO DE DATOS **9**
2.1.1. Diccionario de datos para la Unidad Certificadora 9
2.1.2. Diccionario de datos para la consulta de notas 9

2.2 Scripts de la creación de Base de Datos **10**
2.2.1 Scripts de la Base Datos Para CAOS 10
2.2.1 Scripts de la Base Datos Para Consulta de notas 14

CAPITULO 3 ***15***

3.1 CONFIGURACION DE COMPONENTES **15**
3.1.1 Instalación De White Box 3 15
3.2 Configuraciones de la red 16
3.2.1 Configuración del Hostname 16
3.2.2Configuración del Sysconfig 16
3.2.3 Configuración de las tarjetas de Red 17
3.3. Comprobar la Red 18
3.4. Configuración de los Servicios 18
3.4.1. Configuración del DNS 18
3.4.1.1. Comprobar el DNS 21
3.4.2. Configuración de Sendmail 22
3.4.3. Configuración del Dovecot 24

3.4.5. Configuración del Apache 24
3.4.4. Configuración del Openssl 25

CAPÍTULO 4 *26*

4.1 INSTALACION DE SOFTWARE REQUERIDO **26**
4.1.1 Descargar JDK para Linux 26
4.1.2 Instalar JDK para Linux 26
4.1.2.1. Configuración 26
4.1.2.2. Pruebas 27
4.2. Instalación del Tomcat 28
4.2.1. Configuración 28
4.2.2. Pruebas 29
4.3. Instalación de MySql 29

CAPÍTULO 5 *30*

5.1 CONFIGURACION DEL SISTEMA CAOS **30**
5.1.1. Cofiguracion Del Keystore 30
5.2. Configuración del Tomcat 30
5.3. Configurar Eclipse 31

CAPITULO 6 *34*

6.1 CODIGO **34**
6.1 Codigo de conexion 34
6.2. Codigo para encriptar y desencriptar Clave 35
6.3. Codigo para crear el Certificado 38

MANUAL DE USUARIO

MANUAL DE USUARIO *43*

INTRODUCCIÓN *44*

CAPITULO 1 *45*

1.1 USO DEL SISTEMA CAOS **45**

1.2. USO DE LA PAGINA WEB DE CONSULTA DE NOTAS **50**

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

"Diseño e implementación de una Unidad Certificadora Raíz"

Manual Técnico

Manual Usuario

III SEMINARIO DE GRADUACIÓN

PARALELO "A"

GRUPO #2

Autores:

Katherine Isabel Medina Espinoza

Diana Annabell Vinueza Quimiz

Manuel Geovanny Durán Alvear